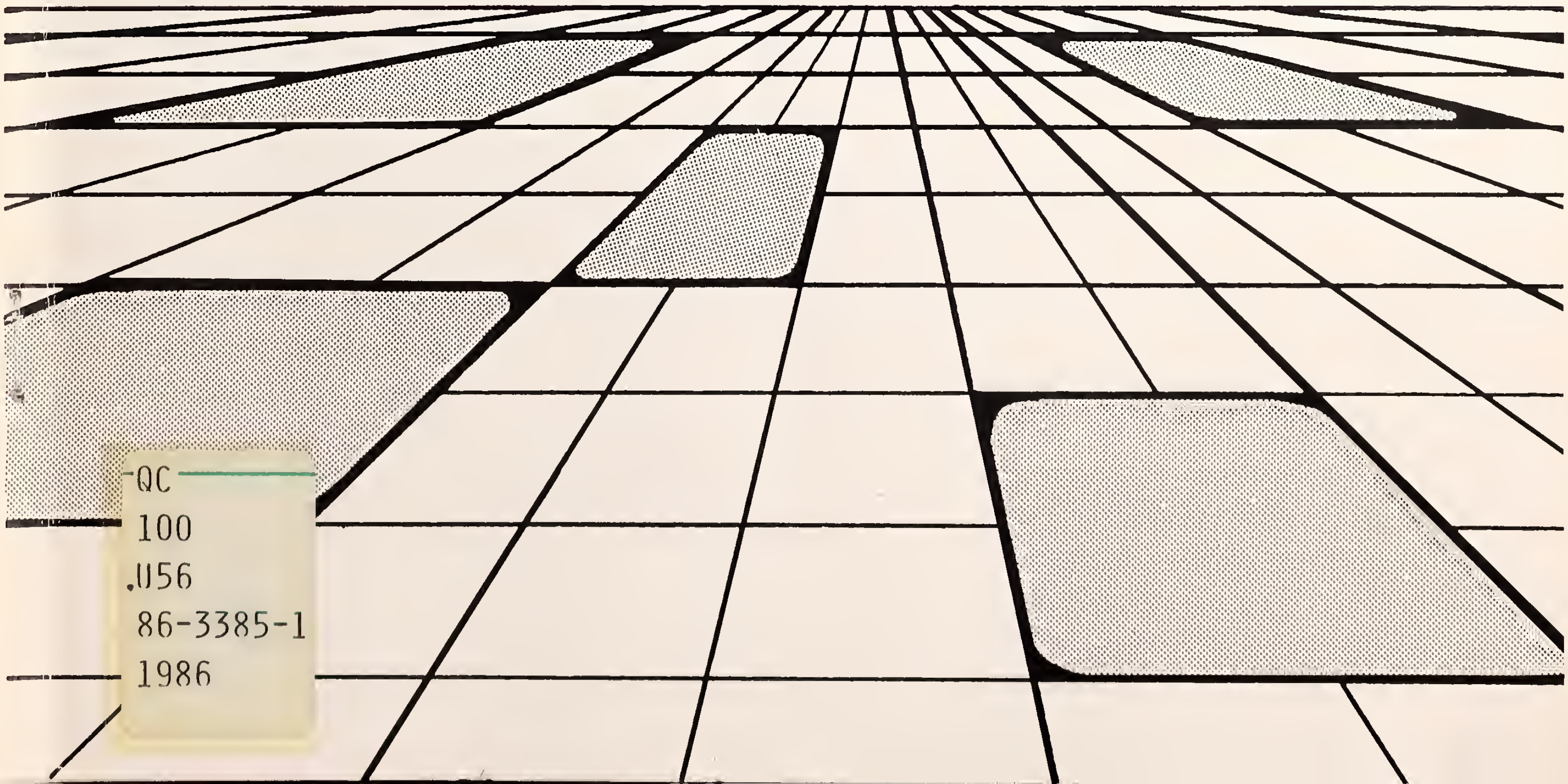
U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards

NBSIR 86-3385-1

Implementation Agreements for Open Systems Interconnection Protocols

NBS Workshop for Implementors of Open Systems Interconnection

Revised July 24, 1986



U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, *Secretary*
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Director*

NBSIR 86-3385-1

Implementation Agreements for Open Systems Interconnection Protocols

NBS Workshop
for Implementors of
Open Systems Interconnection

Revised July 24, 1986

08/13/86

To Whom it May Concern:

This is to advise you that the manuscript, Implementation Agreements Among Implementors of OSI Protocols has been approved by WERB and assigned an NBSIR number 86-3385-1. If there are any questions concerning this note as the approval, please call WERB office.

Rolfe MacCullough

Rolfe MacCullough
WERB Secretary
X2058

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. GENERAL INFORMATION

1.1 PURPOSE OF THIS DOCUMENT

This document records current agreements on implementation details of OSI protocols among the organizations participating in the NBS/OSI Workshop Series for Implementors of OSI Protocols. These decisions are documented to facilitate organizations in their understanding of the status of agreements. This is a standing document that is intended to be updated after each workshop (about every 2-1/2 months).

1.2 PURPOSE OF THE WORKSHOPS

In February, 1983 NBS organized the above named workshops to bring together future users and potential suppliers of OSI protocols. The workshop accepts as input the specifications of emerging standards for protocols and produces as output agreements on the implementation and testing particulars of these protocols. This process is expected to expedite the development of OSI protocols and promote interoperability of independently manufactured data communications equipment.

1.3 RELATIONSHIPS OF WORKSHOPS TO OUTGROWING EVENTS AND ACTIVITIES

The workshops are held for those organizations expressing an interest in implementing OSI protocols. However, there is no corporate commitment to implementations associated with workshop participation. Other events stem from the workshops to which commitments are attached. Sixteen organizations did make formal commitments to implement and demonstrate some of the protocols at the 1984 National Computer Conference. Commitments were made by General Motors, Boeing Computer Services, and 21 vendors to demonstrate, at AUTOFACT 1985, a set of OSI protocols known as the Manufacturing Automation Protocols or MAP specification. This event was an outgrowth of the workshops in that the implementation decisions reached in the workshops were used for the AUTOFACT demonstration. However, the AUTOFACT demonstration was planned and carried out by GM, BCS, and suppliers they selected. This event had no further affiliation with the workshops.

A different activity, initiated by NBS, is the OSINET, which is to be a long standing, globally distributed network that is put in place for purposes such as test methods development and testing of prototype implementations. Presently, 25 organizations have committed to participate. As with the AUTOFACT demonstration, protocols used on OSINET are those agreed to in the workshop. Also, as with the AUTOFACT demonstration, the OSINET has no other affiliation with the workshops. Unlike the AUTOFACT demonstration, however, OSINET participation is open to any organization committing to the OSINET agreements.

1.4 RELATIONSHIP OF THE WORKSHOPS TO THE NBS LABORATORIES

As resources permit, NBS, with voluntary assistance from industry, develops formal protocol specifications, reference implementations, tests and test systems for the protocols agreed to in the workshops. This is work made available to the industry volunteers and to others making valid commitments to organized events and activities such as NCC, AUTOFACT, and OSINET. As soon as this work can be adequately documented it is placed in the public domain through submission to the National·Technical Information Service. Any organization may then obtain the work at nominal charge.

The NBS laboratories bear no other relationship to the workshops.

1.5 STRUCTURE AND OPERATION OF THE WORKSHOPS

1.5.1 Plenary

The main body of the workshops is a plenary assembly. Any organization may participate. Representation is international. NBS prefers for the business of workshops to be conducted informally, since there are no corresponding formal commitments within the workshops by participants to implement the decisions reached. The guidelines we follow are: 1) one vote per company or independent division, 2) only companies that regularly attend should vote, 3) only companies that plan to sell or buy a protocol should vote on its implementation decisions, 4) only companies knowledgeable of the issues should vote, and 5) no proxy votes are admissible.

1.5.2 Special Interest Groups

Within the workshops there are Special Interest Groups (SIGs). The SIGs receive their instructions for their technical program of work from the plenary. The SIGs meet independently, usually during the three day workshop. As technical work is completed by a SIG, it is presented to the plenary for disposition. Companies participating in a SIG are expected to participate in the plenary. Voting rules for SIGS are the same as voting rules for the plenary.

Special Interest Groups sometimes correspond with organizations performing related work, such as ANSI committees. Such correspondence should be sent through the plenary to the parent committee, such as ANSC X3T5 or ANSC X3S3. When SIG meetings take place between workshops, the correspondence from these meetings should be addressed directly to the parent committee and copied to the workshop plenary.

Following are procedures for cooperative work among special interest groups.

a) Any SIG (SIG 1) or individual having issues to discuss with or requirements of another SIG (SIG 2) should bring the matter to the attention of the chairperson of that SIG (SIG 2).

b) The SIG 2 chairperson should bring the matter before SIG 2 for action.

c) SIG 2 should respond to the concerns or needs of SIG 1 or the individual in a timely manner.

d) If the matter cannot be satisfactorily resolved or if the request is outside the charter assigned to SIG 1, then it should be brought before the plenary.

e) SIGs are expected to complete work in a timely manner and bring the results before the plenary for disposition. However, the plenary may elect to act on any issue within the scope of the workshop at any time.

Following are the current charters of the the six Special Interest Groups.

FTAM SIG

Develop phase 2 product-level specifications for FTAM as requested by the Corporation for Open Systems.

Future new work items will be defined in a phase 3 specification. It will contain only extensions of phase 2 FTAM. It is a goal that phase 3 will be backward compatible with phase 2 FTAM. The set of future work items listed below may be changed by the plenary if the work is more appropriate for other SIGs.

High priority work items:

- Develop implementation specifications for ISO CASE and ISO Presentation protocols

- Clean up section 10 of this document

- Specify Reliable File Service

- Specify Recovery and Restart Data Transfer functional units in the user correctable file service

- Specify concurrency control parameter.

Low priority work items:

- Add new document types/constraint sets

- Define subset of authorization requirements

- Specify Presentation Context Management functional unit.

X.400 SIG

Develop product-level specifications for Message Handling Systems using the CCITT X.400 Recommendations as requested by the Corporation for Open Systems.

Develop abstract tests for X.400, as requested by the ad hoc rapporteur for this study question in CCITT. This work is to be submitted by the plenary (after its approval) to The U.S. Department of State as a proposed U.S. contribution to CCITT Study Group VII.

Lower Layer SIG

The Lower Layer SIG will study OSI layers 1-4 and produce recommendations for implementations to support the projects undertaken by the workshops and the work of the other SIGs. Both connectionless and connection-oriented modes of operation will be studied. The SIG will accept direction from the plenary for work undertaken and the priority which it is assigned.

The objectives of the Lower Layer SIG are:

- o Study OSI layers 1-4 as directed by the plenary,
- o Produce and maintain recommendations for implementation of these layers, and
- o Where necessary, provide input to the relevant standards bodies concerning layers 1-4, in the proper manner.
- o Begin work on the implementation specification of the ISO Network Layer Routing Exchange Protocol prior to the ISO draft achieving DIS status.

Performance SIG

The plenary will provide the following inputs to the OSI Performance SIG:

- o the set of applications for which the performance of OSI protocols is of particular concern,
- o the requirements for each application including:
  - performance targets
  - network topology
  - background network loads
  - application traffic characteristics, and
- o this document, "Implementation Agreements Among Implementors of OSI Protocols".

The objectives of the OSI Performance SIG are to:

- determine whether the OSI protocols are able to meet these performance requirements,

- report these determinations to the plenary, and

- where appropriate, provide input to the voluntary stanards bodies concerning changes to existing standards and the requirements for new ones, in the appropriate form.

OSI Security Architecture SIG

GOAL: To develop an overall OSI Security Architecture which is consistent with the OSI and which economically satisfies the primary security needs of both the commercial and Government sectors.

APPROACH: To define a security architecture encompassing the security addenda presently being specified at certain OSI layers, the required cryptographic algorithms and related key management functions, and the security management functions which must be performed between the layers and the peer entitities defined in the OSI architecture.

Directory Services SIG

The DS SIG shall develop implementation specifications, at the request of The U.S. Navy, for directory user agent to directory systems agent and for directory systems agent to directory systems agent. These specifications shall be based upon the current work of CCITT and ISO in this area.

1.6 POINTS OF CONTACT

OSI Workshop - General	John Heafner, NBS, 301/921-3537
OSI Workshop - Registration	Mary Lou Fahey, NBS, 301/921-3516 Kim Brink, NBS, 301/921-3537
FTAM SIG	Rick Peterson, GM, 313/492-6705
X.400 SIG	John Stidd, Xerox, 415/496-6527
Lower Layers SIG	Kevin Miles, DEC, +44-734-868711
Performance SIG	Mary Jane Strohl, CDS, 617/460-0808
Security SIG	Denny Branstad, NBS, 301/921-3427
DS SIG	J. J. Cinecoe, WANG, 617/967-5514
MAP	Gary Workman, GM, 313/575-0632
TOP	Laurie Bride, BCS, 206/763-5719
OSINET	
Steering Committee	Bob Blanc, NBS, 301/921-3817
Technical Committee	Andy Poupart, Tandem, 408/725-6480
SME (MAP/TOP Sponsorship)	Mark Shaw, 313/271-1500 Paul Borawski, 313/271-1500

2. THE PROTOCOLS

The selected protocols for which implementation agreements have been made and are being developed are:

- IEEE 802.2 Logical Link Control
- IEEE 802.3 CSMA/CD Access Method
- IEEE 802.4 Token Bus Access Method
- CCITT Recommendation X.25
- Private Subnetworks
- Network Dependent Convergence Sublayer Protocol between X.25 and ISO Connectionless IP
- X.25 Packet Layer Protocol to support the Connection Oriented Network Service
- ISO Connectionless Internetwork Protocol
- ISO Transport Classes 4 and 0 and Connectionless Protocols
- ISO Session Protocol
- ISO File Transfer, Access and Management Protocol
- ISO Presentation Layer Protocol
- ISO CASE Protocol
- CCITT 400 Series Recommendations for Message Handling Facility
- CCITT and ISO Directory Services Protocol

Reference documents and sources for obtaining them are given under REFERENCES.

3. LOCAL AREA NETWORKS

3.1 IEEE 802.2 LOGICAL LINK CONTROL

The following decisions have been reached with respect to this protocol.

1. Link Service Access Point (LSAP)

The IEEE 802 committee has assigned the code below to address systems using ISO IS 8473 connectionless network protocols. Note that bit zero is transmitted first.

The most significant bit is bit 7, thus this bit pattern represents hexadecimal FE.

0	1	2	3	4	5	6	7
0	1	1	1	1	1	1	1

Fig. 3.1 LSAP bit pattern

2. Type and Class

Only the connectionless type 1, class 1 IEEE 802 link service will be used.

3. Exchange Identification and Test

The Exchange Identification (XID) and Test (TEST) will not be sent. If one is received it will be discarded.

3.2 IEEE 802.3 CSMA/CD ACCESS METHOD

The following decision has been reached with respect to this protocol.

1. Addressing

The 48 bit addressing will be used with the 10 megabit/second baseband coaxial cable specification.

3.3 IEEE 802.4 TOKEN BUS ACCESS METHOD

The workshop participants agreed to inclusion of token bus. The following options were agreed to with respect to Draft F. An asterisk means that the option has been approved. The absence of an asterisk means that the option has not been approved.

1. Repeaters		
	Active Regenerative	
2. Medium		
	Single Cable Coax	*
	Dual Cable Coax	
3. Trunk Cable		
	RG-6	*
	RG-11	*
	Semi-rigid	*
	Other 75 ohm cables	*
4. Trunk Connection Unit		
	75 ohm tee connector	
	75 ohm nondirectional passive impedence-matching tap	

75 ohm directional passive impedence-matching tap	*
5. Transmit Carrier Frequency	
RF	*
Baseband	
6. Modulation	
Phase Continuous FSK	
Phase Coherent FSK	
AM/PSK	*
7. Encoding	
Manchester	
Duobinary	*
8. Data Rate	
1 Mb	
5 Mb	*
10 Mb	*
9. Addressing	
2 octet	
6 octet	*
10. Connector at Station	
50 ohm Male BNC Series	
75 ohm Female F Series	*
11. Priority (4 levels)	*
12. Group Addressing	*
13. Station Management	
14. Broadband Channel Assignments	

Forward	Reverse	
P	3'	*
Q	4'	*
R	4M'	*
S	5'	*
T	6'	*
U	FM1'	*

4. WIDE AREA NETWORKS

4.1 CCITT RECOMMENDATION X.25

When providing CONS, it has been agreed to use X.25 as the standard wide area network protocol. Elements of X.25 are explained in section 6.2.2.

5. PRIVATE SUBNETWORKS

5.1 PRIVATE SUBNETWORKS

The architectures agreed upon allow the use of private subnetworks in addition to private X.25 subnetworks. No particular private subnetwork has been discussed.

6. NETWORK LAYER

6.1 CONNECTIONLESS NETWORK SERVICE (CLNS)

6.1.1 Provisions of CLNS using CLNP (IS 8473)

The following decisions have been reached with respect to this protocol.

1. The "Protocol for providing the connection-less service," ISO IS 8473, will be used to provide the CLNS.

2. The full conformance protocol will be used for concatenated networking. The inactive subset may be used when communicating on a single subnetwork.† The inactive subset is really the absence of an IP and is so indicated by an IP header of one octet of zeros. The nonsegmenting subset will not be supported.

6.1.2 Agreements on Protocol Functions

1. For purposes of demonstration the value to be used to bound the maximum lifetime of the internetwork protocol data unit is three times the network span. The span of the network is considered to be the number of intermediate systems between source and destination plus the destination end system.

2. For purposes of testing, intermediate and end systems will log the following conditions:

   o Discarded protocol data units,
   o Error protocol data units (recorded by system generating the error PDU), and
   o Detection of protocol data units containing unsupported type 3 options.

6.1.3 Agreements on Optional Protocol Functions

1. The security parameter will not be used.

2. Intermediate systems should recognize and support both complete and partial source routing.

   Although end systems may implement source routing (of either type), any requirement for end system source routing is deferred for future study, since support for it may be related to layer management protocols.

† When using the inactive subset, the NSAP Selector must be absent, no segmenting is required, and the AFI and subnetwork number must be the same, i.e., denoting a single subnetwork attachment.

3. Partial source routing will be supported by intermediate systems. The destination should log the route, if possible.

4. The ISO specification will be followed with respect to quality of service.

5. For purposes of testing, checksums will be turned on and used. In operation this is a local decision.

6.1.4 Network Dependent Convergence Sublayer Function (CLNS Over X.25)

A network dependent convergence sublayer protocol operating between CCITT Recommendation X.25 and the ISO Connectionless IP has been agreed to. It shall adhere to the following.

1. Follow ISO 8473 DAD1 (N3601) Working Paper for the SNDCF framework.

2. Only initiate one SVC for all outbound PDUs to any other given intermediate system or PDN end system. Note: this results in at most two SVCs between any pair of systems.

3. Open a connection upon demand.

4. Disconnect SVCs by administrative request and a finite timeout timer.

5. Use the default throughput class from APS.

6. Do not use D-bit or Q-bit.

7. Negotiate window size according to ISO IS 8208. A default of window size two must be supported.

8. Negotiate packet size according to ISO IS 8208. A default of 128 octets/packet must be supported.

9. The SNDCF will "advertise" support for SNSDUs up to 1K via the X.25 M-bit facility.

10. No statistics gathering is required beyond that which the CLNS may gather for the X.25 packet layer or for SNDCF entities.

6.2 CONNECTION MODE NETWORK SERVICE (CONS)

There is interest among a limited set of participants in implementation of the connection mode network service. This section records the agreement of the workshop on the provision of a connection mode network service.

6.2.1 Introduction

The X.25 Packet Level Protocol (PLP) is to be used to provide of the OSI Connection-mode Network Service (CONS). This proposal is independent of lower and higher layer protocol considerations, which are also discussed briefly herein.

When providing the CONS, the following shall apply:

a. the definition of the CONS is as specified in ISO 8348 ("Network Service Definition");

b. the mapping of the elements of the CONS to the elements of the X.25/PLP is as specified in ISO 8878 ("Use of X.25 to Provide the OSI Connection-mode Network Service"); and

c. the general procedures and formats of the X.25/PLP are as specified in ISO 8208 ("X.25 Packet Level Protocol for Data Terminal Equipment"); it should be noted that this standard provides for the use of the X.25/PLP in environments in addition to an X.25 PSPDN, such as point-to-point topologies as well as LANs. (The details of how to use the X.25/PLP in LANs are given in ISO 8881, "Information Processing Systems - Data Communications - Use Of The X.25 Packet Level Protocol in Local Area Networks.")

6.2.2. Provision of CONS Using X.25/PLP

The provision of the CONS using the X.25/PLP is as described in (b) above. This section provides a brief description taken directly from ISO 8878; for more detail, see ISO 8878.

It should also be noted that the X.25/PLP-1984 is capable of supporting the full CONS, including quality-of-service (QOS) aspects. The X.25/PLP-1984 shall be used in LANS and in packet-switched networks allowing the use of the elements of the X.25/PLP-1984 needed to support the CONS. In other packet-switched-network environments, a Subnetwork Dependent Convergence Protocol (SNDCP) shall be used in conjunction with the X.25/PLP-1980 (see section 6.2.5).

6.2.2.1 Overview

6.2.2.1.1 Elements of the X.25/PLP for Support of the CONS

The table below lists the packets and associated fields used when supporting the OSI CONS.

Table 6.1

Packets and fields of the X.25/PLP used to support the OSI CONS

PACKET TYPES[1]	FIELDS[2]
CALL REQUEST INCOMING CALL CALL ACCEPTED CALL CONNECTED	Facility Field, Call and Called User Data Field
CLEAR REQUEST CLEAR INDICATION	Clearing Cause Code Field, Diagnostic Code Field, Facility Field, Clear User Data Field
DATA	M-Bit, User Data Field
RESET REQUEST RESET INDICATION	Resetting Cause Code Field, Diagnostic Code Field
RESTART INDICATION	Restarting Cause Code Field, Diagnostic Code Field

Notes:

1. The packets shown in the table are used in support of the primitives of the OSI CONS. Other packets not shown in the table (i.e., CLEAR CONFIRMATION, RESET CONFIRMATION, and RESTART CONFIRMATION packets) are essential to the use of the packets shown. Yet other packets (i.e., RESTART REQUEST, DIAGNOSTIC, REGISTRATION REQUEST, AND REGISTRATION CONFIRMATION packets) have no relationship to the provision of the OSI CONS.

   In line with these agreements, the INTERRRUPT, INTERRUPT CONFIRMATION, RECEIVE READY, RECEIVE NOT READY, AND REJECT packets are not needed in support of the OSI CONS because the corresponding aspects of the OSI CONS are not to be used (i.e., the Expedited Data Transfer Service and the Receipt Confirmation Service are not used). However, the RECEIVE READY and RECEIVE NOT READY packets are needed for the proper operation of the X.25/PLP.

2. The information in the fields shown in the table have a direct relationship to the parameters associated with the primitives of the OSI CONS. Other fields not shown in the table (e.g., the Logical Channel Identifier, the Packet Type Identifier, the Address Length Fields and the Facility Length Field) are essential to the use of the appropriate packets.

In addition, it is also necessary for the following optional user facilities and CCITT-Specified DTE Facilities to be used and/or agreed to:

a. optional user facilities --

- o Fast Select (facility used)
- o Fast Select Acceptance (facility agreed to, if operating in a packet-switched network environment)
- o Throughput Class Negotiation (facility agreed to and used), and
- o Transit Delay Selection And Indication (facility used);

Note: When operating in a DTE-to-DTE environment without an intervening packet-switched network, the use of the Fast Select Facility Must also be agreed to by the two DTEs. In addition, the Fast Select Acceptance Facility does not apply.

b. CCITT-Specified DTE facilities--

- o Called Address Extension (facility used),
- o Calling Address Extension (facility used),
- o End-to-End Transit Delay Negotiation (facility used),
- o Expedited Data Negotiation (facility used), and
- o Minimum Throughput Class Negotiation (facility used).

Elements of the X.25/PLP not needed in support of the CONS are:

a. Q-bit
b. Permanent Virtual Circuits (PVCs);
c. Diagnostic packets; and
d. optional user facilities other than those listed above.

Elements of the X.25/PLP not used within the scope of these agreements are:

a. D-bit and
b. INTERRRUPT packets

6.2.2.1.2 <u>General Operation of the X.25/PLP-1984 for Supporting the OSI CONS</u>

The X.25/PLP can be used to provide the OSI CONS in an end system connected to a public or private X.25 packet-switched network environment. It can also be used in environments where the end-system is connected to a Local Area Network or where end systems are connected by a dedicated path or by a circuit-switched connection.

The NS provider (more particularly, the Network Layer (NL) entity in an end-system) must provide a translation between

a. the primitives and parameters of the OSI CONS; and
b. the packets and associated fields of the X.25/PLP.

Request and response primitives are translated into packets to be transmitted across the DTE/DXE interface by the NL entity. Received packets, where appropriate, are translated by the NL entity into indication and confirm primitives. These translations are shown in sections 6.2.2.2 through 6.2.2.7.

Note: The Network Service Definition specifies valid sequences of primitives at a NC endpoint and valid parameter responses at the called NC endpoint to Receipt Confirmation negotiation, Expedited Data negotiation, and QOS parameter negotiation. The necessity for the NL entity to police compliance and the NL entity actions to be taken on non-compliance are a local matter and not subject to standardization.

There is also a relationship between some local mechanism used to identify a particular Network Connection (NC) and a Logical Channel (LC) number used to identify a particular virtual circuit. This relationship is a local matter and is not discussed here.

6.2.2.2 Network Connection Establishment Phase

Table 6.2 shows the relationships between the primitives/parameters used during the Network Connection Establishment Phase and the packets/fields associated with the Call Setup Procedures.

Table 6.2

CONS: X.25/PLP mapping for the network connection establishment phase

CONS	X.25/PLP-1984
PRIMITIVES:	PACKETS:
N-CONNECT request	CALL REQUEST
N-CONNECT indication	INCOMING CALL
N-CONNECT response	CALL ACCEPTED
N-CONNECT confirm	CALL CONNECTED
PARAMETERS:	FIELDS (INCLUDING FACILITIES):
Called Address	Called Address Extension Facility
Calling Address	Calling Address Extension Facility
Responding Address	Called Address Extension Facility
Receipt Confirmation Selection	See Note 1
Expedited Data Selection	See Note 1
QOS Parameter Set	Throughput Class Negotiation Facility[2] Minimum Throughput Class Negotiation Facility Transit Delay Selection And Indication Facility End-to-End Transit Delay Negotiation Facility
NS-User-Data	Call and Called User Data Field Fast Select Facility[3]

Notes to Table 6.2:

1. Within the scope of these agreements, the Receipt Confirmation Service and Expedited Data Service will not be used. Therefore, the Network Service provider will indicate the unavailability of these services even if requested by the Network Service user. To do this does not require an explicit protocol mechanism.

2. For proper operation, this optional user facility must also be agreed to for use on the interface.

3. For proper operation, the Fact Select Acceptance Facility must also be agreed to on the interface when accessing a packet-switched network.

6.2.2.3 Network Connection Release Phase

Table 6.3 shows the relationships between the primitives/parameters used during the Network Connection Release Phase and the packets/fields associated with the Call Clearing Procedures.

Table 6.3
CONS: X.25/PLP mapping for the network connection release phase

CONS	X.25/PLP-1984
PRIMITIVES:	PACKETS:
N-DISCONNECT request	CLEAR REQUEST
N-DISCONNECT indication	CLEAR INDICATION, RESTART INDICATION[1], CLEAR REQUEST[2]
PARAMETERS:	FIELDS (INCLUDING FACILITIES):
Originator and Reason	Cause Code and Diagnostic Code Fields[3]
NS-User-Data	Clear User Data
Responding Address	Called Address Extension Facility

Notes to Table 6.3:

1. Receipt of a RESTART INDICATION packet should be treated as receipt of a CLEAR INDICATION packet for every logical channel and then mapped to an N-DISCONNECT indication primitive for every active NC associated with the Packet Level Protocol being restarted. The Restarting Cause Code and Diagnostic Code Fields are then treated in the same manner as the Clearing Cause Code and Diagnostic Code Fields.

2. Used only when the NL entity in the end system originates an N-DISCONNECT indication primitive.

3. The combination of Cause Code and Diagnostic Fields is mapped to/from the combination of Originator and Reason parameters. Where the IS 8208 diagnostic codes are not provided, all Cause/Diagnostic code combinations can be mapped to the Orignator/Reason parameter values of "Undefined".

6.2.2.4 Data Transfer Phase -- Data Transfer Service

Table 6.4 shows the relationships between the primitives/parameters used for the Data Transfer Service and the packets/fields associated with the Data Transfer Procedures.

Table 6.4
CONS: X.25/PLP mapping for the data transfer service

CONS	X.25/PLP-1984
PRIMITIVES:	PACKETS:
N-DATA request N-DATA indication	DATA DATA
PARAMETERS:	FIELDS:
NS-User-Data Confirmation Request	User Data, M-bit See Note 1

Note to Table 6.4:

1. Since the Receipt Confirmation Service is not to be provided, a Confirmation Request is not valid.

6.2.2.5 Data Transfer Phase -- Receipt Confirmation Service

This service is not provided within the scope of these agreements.

6.2.2.6 Data Transfer Phase -- Expedited Data Transfer Service

This service is not provided within the scope of these agreements.

6.2.2.7 Data Transfer Phase -- Reset Service

Table 6.5 shows the relationships between the primitives/parameters used for the Reset Service and the packets/fields associated with the Reset Procedures.

Table 6.5

CONS: X.25/PLP mapping for the reset service

CONS	X.25/PLP-1984
PRIMITIVES:	PACKETS:
N-RESET request	RESET REQUEST
N-RESET indication	RESET INDICATION, RESET REQUEST[1]
N-RESET response	none
N-RESET confirm	none
PARMETERS:	FIELDS:
Originator and Reason	Cause Code and Diagnostic Code Fields[2]

Notes to Table 6.5:

1. Used only when the NL entity in the end system originates an N-RESET indication primitive.

2. The combination of Cause Code and Diagnostic Code Fields is mapped to/from the combination of Originator and Reason parameters. Where the IS 8208 diagnostic codes are not provided, all Cause/Diagnostic code combinations can be mapped to the Originator/Reason parameter values of "Undefined".

6.2.3. Requirements for Underlying Layer

As cited in IS 8208, the X.25/PLP requires the following of the underlying Layer 2:

a. low duplication rate;
b. low missequencing rate;
c. low undetected bit-error rate; and
d. low loss rate.

When operating in a packet-switched (X.25) network environment, the underlying LAPB protocol ensures these requirements. When operating in a LAN environment, LLC 1 can be used. Considerations for LLC1 are given in DP 8881.

6.2.4. Consideration of OSI Transport Layer Protocol Class

It may be useful to explore the desirability of an alternative to the Transport Class 4 protocol for use over the CONS. See section 7, "Transport."

6.2.5 Subnetwork Dependent Convergence Protocol

In cases where an end system is required not to use the elements of the X.25/PLP-1984 needed to support the OSI CONS (e.g., when operating in a packet-switched network environment that will treat as an error the use of any of the CCITT-Specified DTE facilities), then it shall use a Subnetwork Dependent Convergence Protocol (SNDCP) to provide the necessary elements of the OSI CONS to the Network Service user. These elements involve certain aspects of the Network Connection Establishment Phase and the Network Connection Release Phase.

The SNDCP to be used is referred to as the Alternative Procedures for Network Connection Establishment and Release in DP8878. The procedures for the data transfer phase of the SNDCP, as specified in DP8878, are essentially the same as those in sections 6.2.2.4 through 6.2.2.7 of this document.

6.2.5.1 Network Connection Establishment Phase

The procedures for this phase are as specified in DP8878, Annex A, Section 6.1(b) (the Alternative Network Connection Establishment Procedure). These procedures use:

(a) a CALL REQUEST packet containing (in addition to the packet header, the Address Fields, and other facilities):

- the Throughput Class Negotiation Facility if available; otherwise, the throughput of the virtual circuit must be known a priori (by way of the Default Throughput Classes Assignment Facility, for example), and

- a Call User Data Field containing three octets: a one-octet Protocol ID identifying the X.25/PLP-1980 SNDCP encoded X'84' (as assigned by ISO) and a two-octet Continuation Parameter encoded X'2D00' to indicate the first M-bit sequence (MBS) of DATA packets will contain additional Network Connection (NC) Establishment parameters;

(b) a CALL ACCEPTED packet with necessary information for proper X.25 subnetwork interface operation but without any NC Establishment parameters;

(c) an MBS of one or more DATA packets with the Q-bit set to 1, sent by the originator of the NC Establishment attempt, containing all necessary NC Establishment parameters encoded according to DP8878, Annex A, Section 8.7(c) (this MBS is known as an N-CR message); and

(d) an MBS of one or more DATA packets with the Q-bit set to 1, sent by the recipient of the NC Establishment attempt (assuming NC acceptance), containing all necessary NC Establishment parameters encoded according to DP8878, Annex A, Section 8.7(d) (this MBS is known as an N-CC message); see section 6.2.5.2 for NC refusal.

The transit delay of each subnetwork must be estimated.

Similar to the use of the X.25/PLP-1984m the following items are to be noted:

(a) the Network Service provider will indicate the unavailability, during NC Establishment, of the Receipt Confirmation Service and the Expedited Data Service; therefore, no protocol mechanisms are necessary for the negotiation of these services; and

(b) NSAP Addresses are carried entirely in the Address Extension Fields of the N-CR and N-CC messages.

A Connect Response Timer may be used per DP8878, Annex A, Section 6.8(a).

Figure 6.1 shows the NC Establishment Phase.

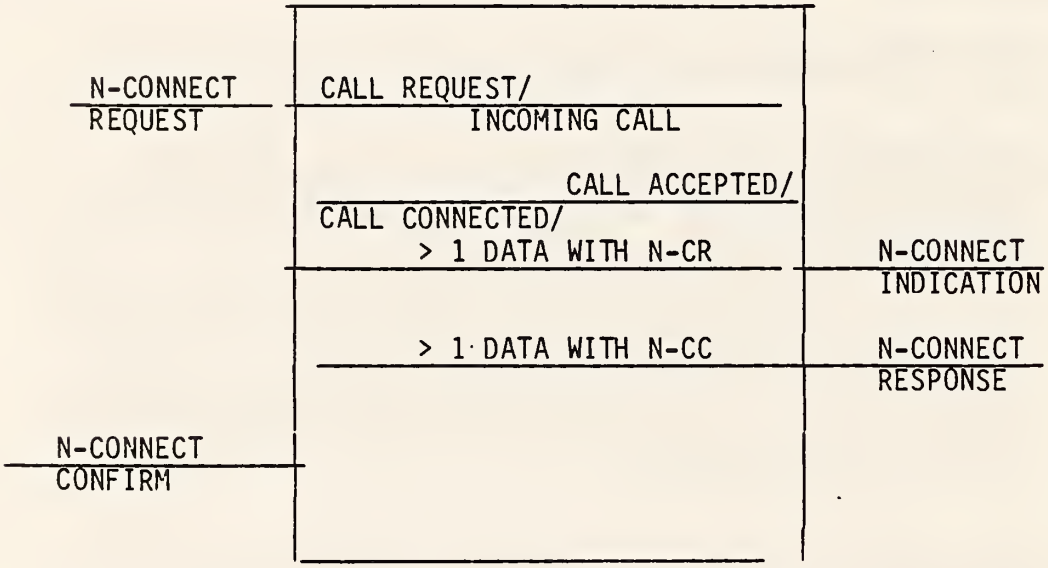


Figure 6.1 Successful NC establishment

6.2.5.2 Network Connection Release Phase

The procedures for this phase are as specified in DP8878, Annex A, Section 6.2 (b) (the Alternative Network Connection Release Procedure). These procedures are also used to refuse an NC establishment attempt. These procedures use:

(a) an MBS of one or more DATA packets with the Q-bit set to 1, sent by the originator of the NC release, containing all necessary NC Release parameters encoded according to DP8878, Annex A, Section 8.8(a) (this MBS is known as an N-DR message and serves as an "invitation to clear" (that is, initiate X.25 clearing procedures to the recipient); and

(b) the X.25 CLEAR REQUEST/INDICATION packet as indicated in DP8878, Annex A, Section 8.8(b).

The above procedures apply to NC release when initiated by the Network Service user. When initiated by the NS provider, the N-DR message will not be sent but the Originator and Reason parameter values shall both be "undefined."

A Disconnect Response Timer may be used per DP8878, Annex A, Section 6.8(b).

Figure 6.2 shows the NC Release Phase.

6.2.5.3 Data Transfer Phase - Data Transfer Service

The Data Transfer Service for the X.25/PLP-1980 SNDCP is the same as for the X.25/PLP-1984 procedures (see section 6.2.2.4 of this document).

6.2.5.4 Data Transfer Phase - Receipt Confirmation Service

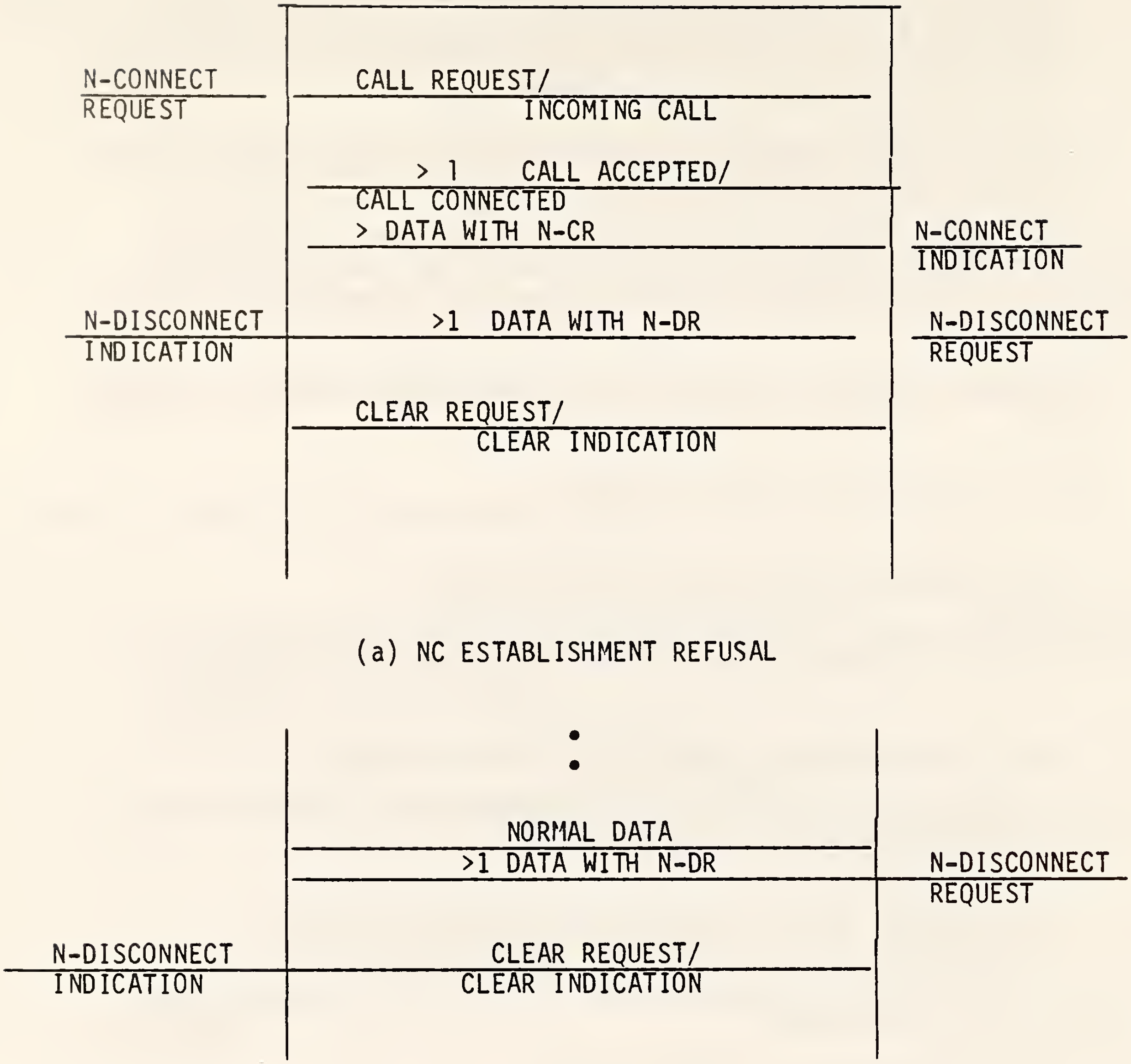
As for the X.25/PLP-1984 procedures, this service is not provided within the scope of these agreements.

6.2.5.5 Data Transfer Phase - Expedited Data Transfer Service

As for the X.25/PLP-1984 procedures, this service is not provided within the scope of these agreements.

6.2.5.6 Data Transfer Phase - Reset Service

The Reset Service for the X.25/PLP-1980 SNDCP is the same as for the X.25/PLP-1984 procedures (see section 6.2.2.7 of this document).



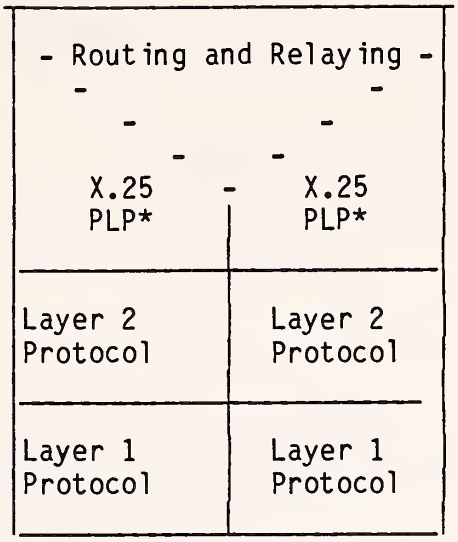
(a) NC ESTABLISHMENT REFUSAL

(b) NC RELEASE AFTER DATA TRANSFER

Figure 6.2 NC release

6.2.6 Interworking

Interworking between subnetworks needs to be specified. The principles of the Internal Organization of the Network Layer, DP 8648, need to be addressed. A generalization of these principles for the OSI CONS case is shown in Figure 6.3.



*This can be either X.25/PLP-1984 or X.25/PLP-1980 with an SNDCP (see Section 6.2.5).

Figure 6.3 Generalized interworking for OSI CONS

The detailed behavior and properties of the Interworking Unit are to be specified in future extensions of this document. DP8881 and DP 8878 discuss various aspects of interworking.

6.3 ADDRESSING AND ROUTING CRITERIA

The following have been agreed to.

1. Do not go beyond ISO routing standardization efforts.

2. Keep the end system as simple as possible. Put the needed compleity into the intermediate systems. Minimize memory overhead and enhance performance in the end systems.

3. Addressing and routing decisions should include the NBS OSINET that permits inclusion of private subnetworks and end systems on PDNs.

4. Permit multiple subnetwork (alternate paths) in a single intermediate system.

5. Allow expandability for additional functionality beyond that required for a specific demonstration.

6. Support a certain minimal topology. Accommodate multiple LANs, WANs, and private subnetworks.

7. Facilitate efficient implementation of intermediate systems. Minimize memory overhead and enhance performance of intermediate systems.

8. The OSINET will accommodate end systems attached directly to PDNs as one requirement.

6.4 GENERAL ADDRESSING AND ROUTING PRINCIPLES

The following have been agreed to.

1. Support DAD2. This supports criterion no. 1.

2. Routing Management

Static Routing:
All end systems and intermediate systems will provide a local mechanism to manipulate (and optionally create) the local routing table. Consistency checking, configuration, and updating of a local table with all other tables will be performed by human operators. The lack of flexibility induced by static procedures can be minimized with the use of the following aids:

- alternate static routes
- route update utility
- FTAM route table distribution
- table format via X.409 (ASN-1) encoding.

Dynamic Routing:
Endsystem-Intermediate System - A standard that provides the functionality of dynamic routing must be provided. The protocol referenced in the "References" section. Item ISO 18, "Network Layer Management Protocol...", provides for this functionality between end and intermediate systems. This standard is expected to be endorsed (and appropriate implementor agreements drafted) to provide this function at a time when it reaches the "Draft Proposal" status.

Intermediate system-Intermediate System

To be determined

3. Routing Principles

The algorithm and data structures used for routing are not specified by this document. Implementors are free to perform these functions in the manner which is most appropriate for their system environment. However, all implementations must have the following characteristics:

a) end-systems

End-systems must recognize a destination address on a directly connected subnet and send the NPDU to the destination system. When an NPDU is destined to a system that is not on a directly connected subnet, the NPDU must be sent to an intermediate system for further routing. The end system may, but is not required to, choose the intermediate system used on the basis of the destination subnet.

b) intermediate systems

For static routing, intermediate systems must recognize all assigned subnetwork addresses and route NPDU's in the following way: If an NPDU is destined to an end system which is connected to the same subnet as the intermediate system, the NPDU is sent directly to the end system. If an NPDU is destined to an end system which is not connected to the same subnet as the intermediate system, the NPDU is sent to an appropriate intermediate system for further routing. If an NPDU is destined to an end system on an unknown subnet, the NPDU is discarded. If the error flag is set, an error NPDU is sent to the source address specified in the errant NPDU.

4. There is a single NSAP selector for each NSAP within an end system. For each end system NSAP, one and only one NSAP address will be used to identify the NSAP. This conforms to the requirements of DAD2 and supports criterion no. 1 to comply with current ISO standards.

5. Hierarchical NSAP addressing should be used to minimize the size of routing tables. This conforms to criterion no. 9.

6. The NSAP address used at the network service interface is based on one and only one subnetwork point of attachment (SNPA). This makes the hierarchial addressing more specific and implies that subnetwork specific addressing information is embedded in the NSAP address.

7. The encoded network protocol address information (NPAI) conveyed in the IPDU representing an NSAP address contains an encoded subnetwork address corresponding to the selected SNPA. This makes hierarchical addressing more specific as applied to addressing formats. The embedding of the subnetwork address facilitates routing and simplifies routing tables.

8. The workshop participants require control over the DSP encoding of the NSAP address.

9. NBS has obtained an IDI with value 4 from ISO for OSINET so that the AFI encoding "47" may be used.

10. Optionally, the NSAP addressing format should be able to support multiple network layer user entities (e.g., transport entities) within one end system.

11. All of the above principles lead to the address format agreed upon and shown below.

47	ISO IDI	NBS ASSIGNED ORGANIZATION I.D.	PORTION OF DSP DEFINED ASSIGNED BY INDIVIDUAL ORGANIZATION ADDRESSING AUTHORITIES
1	2	2	$\leq$ 11

Figure 6.4 NSAP address format

12. If used in the DSP, an NSAP Selector field is not used for routing purposes nor for locating the end system. It only locates the network layer user entity within the end system attached through the addressed NSAP.

13. In order to fulfill the criteria in section 6.3, item 3, a standard that provides the functionality of dynamic routing must be provided. The protocol referenced under ISO, item #18 "Network Layer Management Protocol ..." provides for this functionality between end and intermediate systems. This standard is expected to be endorsed to provide this function at a time when it reaches a "Draft Proposal" status.

7. TRANSPORT

The object of this set of agreements is to support the integration of the types of networks covered here (LANs, packet networks, other WANs) with the smallest possible set of mandatory protocol sets, in accordance with the other agreements already reached. Nothing here shall preclude vendors from implementing protocol suites in addition to the ones described in this document. Two connection oriented transport classes have been identified for implementation (class 0 and class 4). In addition, there is interest among a limited set of participants in implementation of a connectionless transport protocol. Transport class 4 (over CLNP) has been endorsed for general communication between private systems. Transport class 0 (over X.25) is used for communication with public (i.e. PT&T and RPOA) MHS systems operating in accordance with the CCITT X.400 series recommendations. Communicating entities between private MHS systems over an X.25 network can, by negotiation or bilateral agreement, agree to use transport class 0. The connectionless transport protocol can be used with transaction-type implementations.

7.1 TRANSPORT CLASS 4

7.1.1 Transport Class

The following agreement has been reached with respect to this protocol.

Class 4 will be used with the required implementation of the 31 bit sequence space and 16 bit window size. The full protocol will be used including expedited data and negotiation at connection establishment. These two functions were not implemented for the 1984 NCC demonstrations.

7.1.2 Protocol Interpretation

According to the ISO transport specification, a disconnect request is issued in response to a connect request when the maximum number of transport connections is reached or exceeded.

7.1.3 Rules for Negotiation

1. In general, the ISO rules for negotiation will be used, specifics follow.

2. All implementations will send the 16/31 window size/sequence space in the CR TPDU. Implementations must all provide the 16/31 ISO option. Implementations must be able to accept the 4/7 in CR TPDU.

3. The ISO TPDU size is 128 to 8K octets, always negotiated downward. The ISO rules are to be followed, allowing any valid size in the CR TPDU. TPDU size negotiation is a local implementation issue. Each vendor will decide how it is implemented in their end system.

4. The security parameter is optional and user defined in the ISO specification. Implementations should not send the security parameter in the CR TPDU; if received it should be ignored.

5. Both transports must agree to not use checksum, according to the ISO specifications. Requesting its use is an implementation choice. All implementations must be able to operate with checksum if requested.

6. Use of acknowledgement time parameter is optional in ISO 8073. If an implementation is operating any policy which delays the transmission of AK TPDUs, the maximum amount of time by which may single AK TPDU may be delayed shall be indicated to the peer transport service provider using the acknowledgement time parameter. The value transmitted should be expressed in units of milliseconds and rounded up to the nearest whole millisecond.

7. Throughput, priority, and transit delay are optional in the ISO specification. Do not send in the CR TPDU; ignore in the CC TPDU.

8. User data in the CR TPDU and the CC TPDU are optional. No implementation should send; all implementations must be prepared to receive.

7.1.4 Retransmission Timer

It is recommended that the value used for the retransmission timer be based upon the round-trip delay experienced on a transport connection. The implementation should maintain, and continually update, an estimate of the round-trip delay for the TC. From this estimate, a value for the retransmission timer is calculated each time it is started. An example technique for maintaining the estimate and calculating the retransmission timer is described below. Further information on similar techniques may be found in the literature [Edge 84, Jain 85, Mill 83].

The value of the retransmission timer may be calculated according to the following formula:

$$t \leftarrow kE + w$$

In this formula, E is the current estimate of the round-trip delay on the transport connection, w is the value of the acknowledgement time parameter received from the remote transport service provider during connection establishment, and k is some locally administered factor.

A value for k should be chosen to keep the retransmission timer sufficiently small such that lost TPDUs will be detected quickly, but not so small that false alarms are generated causing unnecessary retransmission.

The value of E may be calculated using an exponentially weighted average based upon regular sampling of the interval between transmitting a TPDU and receiving the corresponding acknowledgement. Samples are taken by recording the time of day when a TPDU requiring acknowledgement is transmitted and calculating the difference between this and the time of day when the corresponding acknowledgement is received. New samples are incorporated with the existing average according to the following formula.

$$E \leftarrow E + (1 - \alpha)S$$

In this formula, S is the new sample and alpha is a parameter which can be set to some value between 0 and 1. The value chosen for alpha determines the relative weighting placed upon the current estimate and the new sample. A large value of alpha weights the old estimate more heavily causing it to respond only slowly to variations in the round-trip delay.

A small value weights the new sample more heavily causing a quick response to variations. (Note that setting alpha to 1 will effectively disable the algorithm and result in a constant value for E being that of the initial seed.)

If alpha is set to $1-2^{-n}$ for some value of n, the update can be reduced to a subtract and shift as shown below.

$$E \leftarrow E + 2^{-n}(S - E)$$

When sampling, if an AK TPDU is received which acknowledges multiple DT TPDUs, only a single sample should be taken being the round-trip delay experienced by the most recently transmitted DT TPDU. This attempts to minimise in the sample any delay caused by the remote transport service provider withholding AK TPDUs.

7.1.5 Keep-Alive Function

The Class 4 protocol detects a failed transport connection by use of an 'inactivity timer'. This timer is reset each time a TPDU is received on a connection. If the timer ever expires, the connection is terminated.

The Class 4 protocol maintains an idle connection by periodically transmitting an AK TPDU upon expiration of the 'window timer'. Thus, in a simple implementation, the interval of one transport entity's window timer must be less than that of its peer's inactivity timer, and vice versa. The following agreements permit communicating transport entities to maintain an idle connection without shared information about timer values.

1. In accordance with ISO 8073, clause 12.2.3.9.a, all implementations must respond to the receipt of a duplicate AK TPDU by transmitting an AK TPDU containing the 'flow control confirmation' parameter.

2. Implementations must always transmit duplicate AK TPDUs on expiration of the local window timer (see ISO 8073, clause 12.2.3.8.1). Receipt of this TPDU by the remote transport entity will cause it to respond with an AK TPDU containing the 'flow control confirmation' parameter. When this is received by the local transport entity, it will reset its inactivity timer. See figure 7.1.

3. It is a local matter for an implementation to set the intervals of its timers to appropriate relative values. Specifically:

   a) The window timer must be greater than the round-trip delay. See section 7.1.4.

   b) The inactivity timer must be greater than two times the window timer; and should normally be an even greater multiple if the transport connection is to be resilient to the loss of an AK TPDU.

A duplicate AK TPDU (see Figure 7.1) is one which contains the same values for YR-TU-NR, credit, and subsequence number as the previous AK TPDU transmitted. A duplicate AK TPDU does not acknowledge any new data, nor does it change the credit window.

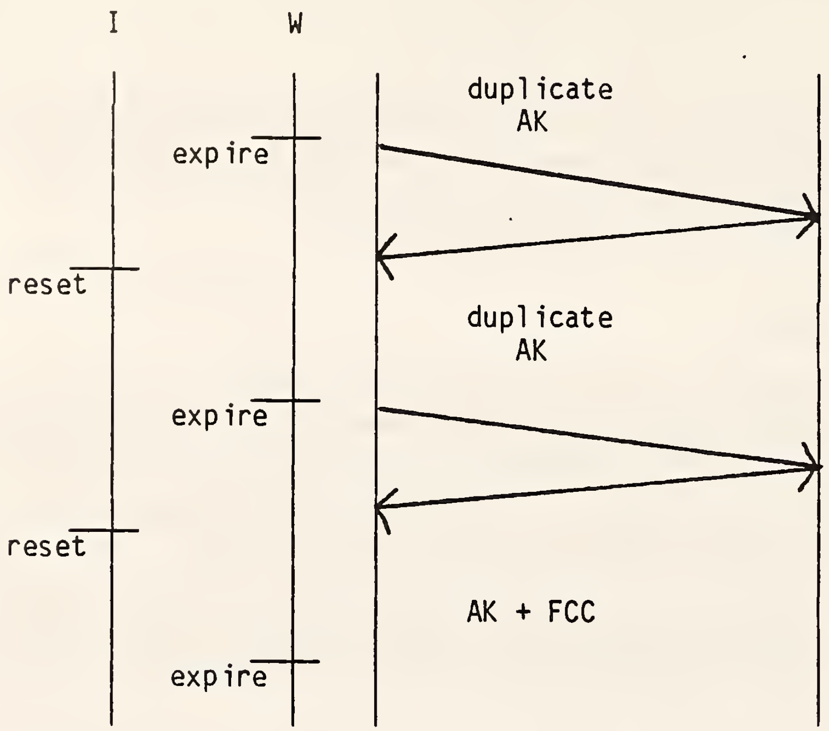


Figure 7.1 AK exchange on idle connection

7.2 TRANSPORT CLASS 0

7.2.1 Transport Class

Transport class 0 over X.25 is mandatory (see X.400) for use in communicating with public MHS systems operating in accordance with the CCITT X.400 series recommendations. The purpose of the agreements concerning transport class 0 is to allow connection to these public services. Transport class 0 over X.25 can also be used in communicating between PRMDs, but it is agreed that Transport Class 4 and CLNP over all types of lower layer networks allows a larger range of interoperation and is recommended.

7.2.2 Protocol Interpretation

Transport class 0 is a relatively simple protocol providing little opportunity for conflicting interpretations. A few relevant agreements follow.

1. The Disconnect Request (DR) TPDU shall be limited to the first seven octets - "LI" plus "fixed part".

2. The Error (ER) TPDU may be used at any time and upon receipt requires that the recipient disconnect the network connection (and by extension the transport connection).

7.2.3 Rules for Negotiation

1. The ISO rules for negotiations will be used.

7.3 CONNECTIONLESS TRANPORT

Document ISO IS 8072/DAD1 is the Transport Service Definition covering Connectionless-mode Transmission. Document ISO DIS 8602 is the Protocol for providing the Connectionless-mode Transport service.

8. SESSION

Session services are defined to meet the needs of many applications. The Session service is defined in ISO 8326 (CCITT X.215) and the session protocol is defined in ISO 8327 (CCITT X.225). The general agreements about the use of session are documented below in section 8.1, followed by agreements that are related to specific applications.

8.1 GENERAL

The services of the Session kernel functional unit are used as specified in the standard.

Session Connection
Data Transfer
Orderly Release
Provider Abort
User Abort

Basic concatenation is required by the session protocol standard. Extended concatenation is not required and can be refused using the normal negotation mechanisms provided by the session protocol.

Session segmenting is not required and can be refused using the normal negotiation mechanisms of the session protocol.

Reuse of a transport connection is not required and can be refused using the normal negotiation mechanisms of the session protocol.

The use of transport expedited is as stated in the session protocol specification. That is, if transport expedited is available it must be used.

The maximum length of the reflect parameter values parameter in the S-P-Exception-Report is 1024 octets.

The user data parameter on S-CONNECT SPDU will allow unlimited length user data as specified in the proposed Session draft addendum. This is due to the fact that the combined P-CONNECT, A-ASSOCIATE, and F-INITIALIZE may exceed the existing 512 byte limit.

8.2 SESSION REQUIREMENTS FOR FTAM

The phase 2 FTAM requires the following functional units in addition to those specified in section 8.1.

Functional Units	Session Services
Duplex	- - - - - -

Note: Implementation of the Resynchronize functional unit is highly recommended since the F-CANCEL service may be ineffective when mapped to S-DATA.

8.3 SESSION REQUIREMENTS FOR MESSAGE HANDLING

The MHS application requires the following functional units in addition to those specified in section 8.1.

Functional Units	Session Services
Exceptions	User Exception Reporting Provider Exception Reporting
Activity Management	Activity Start Activity Resume Activity End Activity Interrupt Activity Discard Please Tokens Give Tokens Give Control
Half-duplex	Give Tokens Please Tokens
Minor Synchronize	Minor Synchronization Point Give Tokens Please Tokens

Note: Restricted use is made by the RTS of the session services implied by functional units selected. Specifically,

- NO use is made of S-TOKEN-GIVE
- S-PLEASE-TOKENS only asks for the data token.

The following additional points should be noted.

- In S-CONNECT, the SynchronizationPointSerial Number should not be present.
- Format of the SessionConnectionID is described in Version 3 of the X.400-Series Implementor's Guide.

9. SERVICE ACCESS POINTS AND SELECTORS

9.1 UPPER LAYER AGREEMENTS

The following upper layer addressing agreements have been reached.

1. The combination of NSAP address, TSAP selector, SSAP selector, PSAP selector and PSAP address must be unique to identify an application entity.

2. It is implicitly agreed that the procedure followed for the assignment of NSAP addresses insures that they are globally unique.

3. The assignment of TSAP, SSAP, and PSAP selectors is a local end system issue and the values are administered locally.

4. SSAP selectors are encoded as a string of octets, the meaning of which is known only to the local system. The length of the string cannot exceed 16 octets.

5. PSAP selectors are encoded as a string of octets, the meaning of which is known only to the local system. The length of the string cannot exceed 16 octets.

9.2 TRANSPORT CLASS 4 SERVICE ACCESS POINTS OR SELECTORS

1. The existing encoding for TSAP selectors in the CR and CC TPDU's will be maintained. The TSAP selector field in these TPDU's shall be encoded as a variable length field, two octets in length, and will be interpreted as a bit string (not an integer).

9.3. TRANSPORT CLASS 0 SERVICE ACCESS POINTS

For communicating with public MHS systems, Section 5 of X.410 specifies the use and format of TSAP identifiers.

10. ISO FILE TRANSFER & ACCESS MANAGEMENT PROTOCOL

The following general agreements have been reached with respect to this protocol.

1. FTAM will be implemented in at least two phases. Phase 1 agreements are contained in section 10.1 and are devised for the MAP2.1 specification. Phase 2 agreements are contained in section 10.2 and are devised at the request of the Corporation for Open Systems for 1987 product implementations.

2. The phase 1 FTAM implementation specification is based on the second ISO draft proposal. The phase 2 FTAM specification is based on the DIS and later will be based on the IS.

10.1 PHASE 1 FTAM IMPLEMENTATION SPECIFICATION

10.1.1 Phase 1 FTAM Services

The following subset of file transfer services has been agreed to.

F - INITIALIZE
F - SELECT
F - OPEN
F - READ, F-WRITE
F - DATA-END
F - TRANSFER-END
F - CLOSE
F - DESELECT
F - U-ABORT, F-P-ABORT
F - TERMINATE

(The implementation of F-CANCEL is optional.)

Both F-READ and F-WRITE will be implemented. There is at most one F-OPEN and at most one F-READ or F-WRITE per file selection; more than one file may be selected sequentially over the lifetime of a connection. A Session connection is established at the beginning of a file activity and terminated when the file transfer connection is released.

The following limited management subset of services has been agreed to.

F - CREATE
F - DELETE
F - READ-ATTRIBUTE

(Note that this subset is complete with respect to the ISO limited file management functional unit.)

The user correctable file service and the storage subset of the virtual filestore attribute group will be implemented.

10.1.2 Phase 1 File Attributes

Phase 1 FTAM implementations will negotiate for the use of Storage Attributes.

The following file attribute agreements have been reached.

1. FILENAME

   DP8571 Part 2 Section 11 minimum range for file name will be supported (1-8 characters). No maximum length or format restrictions. Any system which does not support extended file name characteristics would reject the (select, create, delete) request of such a file name.

   File names will consist of upper case characters and numbers. The first character of a filename may not be a number.

2. ACCESS STRUCTURE TYPE

   The access structure type will be unstructured. Flat and hierarchical are not used.

3. PRESENTATION CONTEXT NAME

   Two structures have been approved.

   VARCRLF: Text format. Each line is variable length and must be terminated by a CRLF pair. Line length is limited to 250 characters plus CRLF. Form feed is excluded. CR and LF cannot occur separately. The ISO 646 character set will be used.

   UNDEF: Bit string encoding. No embedded structure is recognized.

4. CURRENT FILESIZE

   Value represented as conformant with second ISO draft proposal.

5. REQUESTED ACCESS

   This attribute will support read, replace, read attribute, and delete file.

6. CURRENT ACCESS STRUCTURE TYPE

The current access structure type will be unstructured.

7. CURRENT PRESENTATION CONTEXT

This attribute is the same as PRESENTATION CONTEXT, i.e., ASCII or binary.

8. FUTURE FILESIZE

This is a provider option. Value represented as conformant with second ISO draft proposal.

9. DATE AND TIME OF CREATION

This is a provider option. The value is represented as conformant with second ISO draft proposal. Resolution and accuracy are implementation dependent.

10.1.3 ISO Deviations and Selections

1. (DP8571/3 8.1 Table 1) Implementation of the F-CANCEL service for non-recoverable errors in the read and write functional units is optional. An implementation which receives an F-CANCEL indication may issue an F-ABORT.

2. (DP8571/3 8.2 1b) There will be no implementation of the grouping functional unit. The concatenation constraint on establishing and releasing a file open regime is not supported.

3. (DP8571/3 14.1.2.5) F-INITIALIZE The value of the Service Type parameter chosen for implementation is "user correctable service".

4. (DP8571/3 14.1.2.6) F-INITIALIZE The value of the Service Class parameter chosen for implementation is "file transfer class".

5. (DP8571/3 14.1.2.7) F-INITIALIZE The value of the Functional Units parameter chosen for implementation is the set "read", "write" and "limited file management".

6. (DP8571/3 14.1.2.8) F-INITIALIZE The value of the Attribute Groups parameter chosen for implementation is "storage".

7. (DP8571/3 14.1.2.9) F-INITIALIZE The value of the Rollback Availability parameter chosen for implementation is "no rollback".

8. (DP8571/3 14.1.2.11) F-INITIALIZE The values of the Presentation Context Name parameter chosen for implementation are defined in section 10.1.2 of this document.

9. (DP8571/3 17.1.2.2) F-OPEN The values of the Processing Mode parameter chosen for implementation are "read" or "replace".

10. (DP8571/3 17.1.2.3) F-OPEN The Presentation Context parameter must be present to indicate the context chosen for the transfer. (This is due to the lack of a presentation layer implementation at this time.)

11. (DP8571/3 24.3.) F-DATA will be encoded as:
IDENTIFIER: context specific tag value [55]
Length: (Length of OCTETS)
Contents: (octets)
There is an explicit F-DATA PDU because of the direct FTAM/Session mapping.
The ASN.1 definition of F-DATA is:

```
     F-DATA request ::= data[55]IMPLICIT OCTETSTRING
```

12. The ASN.1 type "ISO646String" will be used instead of ASN.1 type "GraphicString" in the abstract syntax. Wherever "GraphicString" is specificed in the abstract syntax, the abstract syntax is modified to specify "ISO646String."

13. The PresentationContext attribute on the SELECT/CREATE request is restricted to a sequence of only one PresentationContextName.

14. PresentationContextName parameters will be encoded as:

```
PresentationContextName   ::= [Application 13] PrintableString.
```

15. The abstract syntax for F-INITIALIZE will be:

```
F-INITIALIZE request ::= SEQUENCE

     protocolId [0] INTEGER{isoFTAM (0)} ,
     versionNumber [1] IMPLICIT SEQUENCE{
          major INTEGER, minor INTEGER},
          --initially {major 0, minor 0}
     serviceType [2] INTEGER{
          reliable (0), userCorrectable (1) ,
     serviceClass [3] INTEGER  transfer (0) ,
          access (1), management (2)},
     functionalUnits [4] BITSTRING{
          read (0), write (1), fileAccess (2),
          limitedFileManagement (3),
          enhancedFileManagement (4),
          grouping (5), recovery (6),
          restartDataTransfer (7)} ,
      attributeGroups [5] BITSTRING{
          storage (0), security (1)},
      rollbackAvailability [6] BOOLEAN DEFAULT FALSE
      identityOfInitiator [7] GraphicString OPTIONAL,
      CurrentAccount OPTIONAL,
      filestorePassword [8] OCTETSTRING OPTIONAL,
      checkpointWindow [9] INTEGER OPTIONAL,
      presentationContextNames [10] IMPLICIT
          SEQUENCE OF PresentationContextName OPTIONAL
```

```
F-INITIALIZEresponse ::= SEQUENCE

          Diagnostic,
          protocolID [0] INTEGER {isoFTAM (0)}
          versionNumber [1] IMPLICIT SEQUENCE{
             major INTEGER, minor INTEGER},
             --initially {major 0, minor 0}
          serviceType [2] INTEGER{
             reliable (0), userCorrectable (1)},
          serviceClass [3] INTEGER {transfer (0),
             access (1), management (2)},
          functionalUnits [4] BITSTRING {
             read (0), write (1), fileAccess (2),
             limitedFileManagement (3),
             enhancedFileManagement (4),
             grouping (5), recovery (6),
             restartDataTransfer (7)},
          attributeGroups [5] BITSTRING{
             storage (0), security (1)}
          rollbackAvailability [6] BOOLEAN DEFAULT FALSE,
          checkpointWindow [7] INTEGER OPTIONAL,
          presentationContextNames [8] IMPLICIT
             SEQUENCE OF PresentationContextName OPTIONAL}
```

16. The PresentationContextName parameter on the F-INITIALIZE request and response will be a sequence of presentation context names. The corrected abstract syntax for this parameter will be:

```
PresentationContextNames [10] IMPLICIT SEQUENCE OF
                        PresentationContextName  OPTIONAL,
```

The value of this parameter, if present, will establish limits for the life of the FTAM association.

17. The PresentationContextName parameter on the F-OPEN request and F-OPEN response will be mandatory instead of optional.

18. Phase 1 FTAM products will use the state machine in the FTAM second ISO draft proposal part 4, except where explicitly specified differently in these agreements.

10.1.4 Further Implementation Details

The following recommendations are not specified by the standard but are required to be agreed upon in order to insure that different implementations work together smoothly.

1. The FTAM implementation is mapped directly to Session services and does not incorporate the use of CASE at this time. The F-INITIALIZE request is embedded in the S-CONNECT request and the F-INITIALIZE response is embedded in the S-CONNECT response, not sent as an S-DATA request after the Session has been established.

   If the F-INITIALIZE response is a positive confirmation, the F-INITIALIZE response is mapped onto S-ACCEPT. Otherwise, is is mapped onto S-REFUSE.

2. F-U-ABORT and F-P-ABORT are mapped to S-DATA requests. The receiver of an F-U-ABORT or an F-P-ABORT must issue an S-U-ABORT request. This is due to a possible S-RELEASE collison should both entries wish to abort simultaneously.

3. Implementations should be able to parse all valid second ISO draft proposal optional parameters if they are present in the PDU. Only those optional parameters specified in the agreements are required to be supported for request and response PDUs.

4. All second ISO draft proposal optional parameters identified in these agreements as mandatory must be supported. If these parameters are not present, their semantics are a local issue and, further, the request should not be refused.

5. Error Handling and Diagnostic

   Action that is permitted by the standard is not restricted in this implementation. Implementations may return a response with a negative diagnostic or issue an F-U-ABORT request.

6. Rollback

   "No rollback" implies that in case of failure, the status of the involved files is unpredictable, i.e., the parties may choose, in their implementations, if they want to leave as is the partially transferred file or cancel it.

   However, in order to avoid dangling conditions, after a failure (i.e., after an F-U-ABORT or F-P-ABORT) it is recommended, whatever the choice, to always put the files in a status which does not prevent further access both from remote and local systems.

7. Concurrency

   No concurrency rules are adopted since the file attributes governing them will not be supported. Therefore, each implementation may choose the degree of concurrency on the local files, as a local matter. However, in order to minimize possible source of errors, the following implementation rules are recommended.

   a. A file may be involved in several transfers simultaneously only if accessed in reading mode.
   b. If a file is involved in a transfer in writing mode, any other request of access to that file (either for Read or Write) should be rejected.

c. A file cannot be modified by local users while it is involved in a transfer operation (either in Read or Write). (As a practical matter, this may be very difficult to implement, regardless of the protection mechanisms provided by the local operating system.)

8. The ISO ASN.1 sytax will be used for encoding of PDU headers.

9. The indefinite length style of ASN.1 encoding is not supported.

10. The maximum PDU length is 1024 octets.

11. If a read attribute request specifies an NBS optional attribute that is not supported by an implementation's virtual filestore, then that attribute will not be returned in the Read Attribute response. It is recommended that an implementation return an appropriate diagnostic in this situation (e.g., 400, "attribute non-existent").

12. In the FTAM second ISO draft proposal, the parameter DIAGNOSTIC is defined as an implicit sequence of diagnostics. The order of the diagnostics is arbitrary and has no significance.

13. If a Responder receives an S-CONNECT indication, and the user data carried on that indication are "corrupted" (i.e., decoding the PCI results in an unrecoverable error), then

    1) If the user data can be identified as an F-INITIALIZE request, the Responder should respond with an F-INITIALIZE response (-) or F-P-ABORT with diagnostic conveying an unrecoverable error. This is to be carried on an S-CONNECT response (-) (i.e., rejecting the connection).

       The Responder should further clarify the error in "further details field of diagnostic".

       Parameters on the F-INITIALIZE response should be reflected where appropriate; where this is not possible, default values should be chosen by the Responder where mandatory parameters are required and cannot be reflected.

    2) If the user data cannot be identified as an F-INITIALIZE request, the Responder should respond with an S-U-ABORT with reason code appropriate to the error.

10.2 PHASE 2 FTAM IMPLEMENTATION SPECIFICATION

10.2.1 Assumptions

1. Implementations will be based on the ISO 8571 DIS version of FTAM. When the IS text is approved following the close of DIS ballots the agreements will be modified as necessary to meet the IS specifications.

2. FTAM Protocol machines must be able to parse and process up to 7K octets of File PCI and FTAM user data (including grouped FPDUs) as they would be encoded with the ASN.1 Basic Encoding Rules. It is recommended, however, that Presentation user data not be restricted in size.

3. In order to maximize interoperability it is important for implementations of FTAM service providers not to restrict unnecessarily the service user's ability to generate arbitrary file service requests, as otherwise they may not be able to work with FTAM Responders whose operation is constrained by their mapping of the FTAM virtual filestore to their local filestore. For example, error procedures should only be invoked at the actual occurence of an error, not at the specification of options which might result in an error.

4. Implementations should be able to parse all valid DIS optional parameters if they are present in the PDU. Only those optional parameters specified as mandatory in these agreements are required to be supported for request and response PDUs. If these parameters are not present, it is a local implementation issue to assign a default value. A request should not be refused if a parameter which is optional in the FTAM standard, but is mandatory in these agreements, is not present.

5. Consideration of any standarized service interface is outside the current work items of the FTAM SIG.

10.2.2 Presentation Agreements

The following Abstract Syntaxes are supported:

- ISO 8571-FTAM
- ISO 8571-FADU
- ISO 8650-ACSE1
- NBS-AS1
- NBS-AS2
- NBS-AS3

A registration mechanism will be determined to provide registered names for these syntaxes.

If the presentation context management functional unit is available, it is possible to use P-ALTER-CONTEXT to negotiate the use of an abstract syntax.

10.2.3 FTAM Service Type Agreements

The user correctable service level (excluding Recovery and RestartDataTransfer functional units) will be implemented. There is no requirement to implement the error recovery protocol machine.

10.2.4 Service Class Agreements

Implementation of the following service classes is defined.

- File Transfer
- File Access
- File Management
- File Transfer and Management
- Unconstrained

10.2.5 Functional Unit Agreements

Implementation of the following functional units is defined.

- Kernel
- Read
- Write
- File Access
- Limited File Management
- Enhanced File Management
- Grouping

10.2.6 File Attribute Agreements

1. A value for file name and contents type will always be available. Only the kernel group of attributes is required.

2. Requests for an attribute value shall always return one of the following:

   (a) An actual file attribute value.

   (b) A value indicating that the attribute value is not available at this time. Optionally a diagnostic may be provided indicating that the attribute is not supported.

The set of file management-related diagnostics will be maintained.

3. If the optional Storage Group is implemented, the following attribute must have an actual value available:

   - Permitted Actions.

4. If the optional Security Group is implemented, the following attribute must have an actual value available:

   - Access Control.

5. Implementation of the private group is not specified.

6. Contents Type attribute is limited to the Document Type Name form.

7. A minimum range for filename values is required (1-8 characters). No maximum length or format restrictions apply. A system which does not support multi-component filenames or extended filename characteristics may reject a request involving such a filename. All systems must be able to interpret single component filenames. Requests using single component filenames will be responded to using single component filenames. Responses to requests involving two or more component filenames are not defined here but may be interpreted via bilateral or other external agreement. Use of filenames with multiple components is discouraged.

10.2.7 Document Type Agreements

The following document types are defined.

```
NBS-1 UNDEF
NBS-2 VARCRLF
NBS-3 8859VARCRLF
NBS-4 TEXT
NBS-5 8859TEXT
NBS-6 SEQUENTIAL
NBS-7 RANDOM
NBS-8 INDEXED
NBS-9 FILE_DIRECTORY
```

Part of the ongoing work of the FTAM SIG is to define, discuss and incorporate other file types. Detailed document type definitions are given in APPENDIX D.

Document type Names:

```
• DTN ::= DTName | DTName params
  DTName ::= OBJECT IDENTIFIER
  params ::=:param | :param params
  param ::= Primtype | PrimType, param

  PrimType := INT - <n*>
            | BIT - <n2>
            | IA5 - <n1>
            | 8859 - <n1>
            | OCT - <n1>
            | UTC
            | GEN
            | NULL
            | BOOL
            | FLOAT - <n3, n4>
```

<n1> - Maximum number of characters/octets in string.

<n2> - Number of bits in string (i.e., nonvarying).

<n3> - The minimum number of bits required to be mantissa
for relative precision.

<n4> - Number of bits required to represent unbiased integer exponent.

<n*> - Number of octets required to represent in 2's complement format
the largest integer to be passed.

The primitive data types and minimal size range which an implementation must accept are given in the following table.

Table 10.1 FTAM primitive data types

PRIMITIVE DATA TYPE		REPRESENTATION IN PARAMETER	MINIMUM RANGE (OCTETS)
ASN.1	INTEGER	INT <N*>	(1 - 2)
ASN.1	Bit String	BIT <N2>	(0 - 1)
ASN.1	IA5String	IA5 <N1>	(0 - 134)
NBS-AS1	8859String	8859 <N1>	(0 - 134)
ASN.1	OCTETSTRING	OCT <N1>	(0 - 512)
ASN.1	BOOLEAN	BOOL	
ASN.1	NULL	NULL	
ASN.1	Generalized Time	GEN	
ASN.1	Universal Time	UTC	
NBS-AS1	Floating Point	FLOAT <N3,N4>	

Note: The primitive data types and their maximum ranges for a specific file as described by the parameters above are maintained in the contents type file attribute. The contents type file attribute value is established at the file's creation and cannot be changed via FTAM for the life of the file. This implies that the data element types and ranges and data unit formats are fixed for all accessors of that file as long as the file exists.

An object identifier is a string of integers; FTAM document type parameterization will be achieved by exploiting that structure.

The final registration authority entity will be followed by a data unit description. The data unit description is a series of data element descriptions. Each data element description is an integer identical to the ASN.1 type code, followed by any required parameter values (as integers).

The following values correspond to the NBS primitives not found in ASN.1 and the integer value for ":" (separator).

FLOAT	- 127
8859	- 126
:	- 125

The following abstract syntax definition is given for the respresentation of floating point numbers. The ability to transfer floating point numbers is not required. The only semantics associated with the floating point primitive type is its use to convey a value of relative precision.

The following notation is believed to allow the movement of the same semantics as existing standards for floating-point numbers (IEC 559 and IEEE 754) as well as represent an infinite number of binary floating point numbers.

```
FloatingPointNumber  ::= [PRIVATE 0]   CHOICE
                  {
                  finite         [0]   IMPLICIT SEQUENCE
                         {
                         Sign,
                         mantissa BIT STRING,
                         exponent    INTEGER
                         }
                  infinity          [1] IMPLICIT Sign,
                  signallingNaN     [2] IMPLICIT NaN,
                  quiteNaN          [3] IMPLICIT NaN,
                  zero              [4] IMPLICIT NULL}
Sign                                ::=       INTEGER
                  {
                  positive          (0)
                  negative          (1)
                  }
NaN                                 ::=       INTEGER
```

Notes: 1. The mantissa is a number in the range 1/2 ≤ mantissa < 1.
2. Mantissa * $2^{exponent}$.
3. The first bit in the mantissa is most significant.
4. See IEEE 754 for definitions of terminology such as NaN.

10.2.7.1 Character Sets

The character sets IA5 and 8859/1 have been specified for use. The following describe how each of the character sets will be implemented.

1. IA5

The IA5 character set leaves 2 options and 10 characters unspecified. The following definitions will be used.

2/3	#
2/4	$
4/0	@
5/11	[
5/12	\
5/13	]
5/14	^
6/0	`
7/11	{
7/12	|
7/13	}
7/14	~

(Note this is exactly the International Reference Version (IRV) specified in the IA5 standard except that the code 2/4 has the graphic rendition "$" instead of the IRV-specified value of "¤")

The following details how control characters should be handled.

a) The semantics of format effectors will be preserved.

b) Transmission control characters, device control characters, information separators, and "other" control characters will simply be preserved via their codes.

c) Code extension should not be used. If it is, the code extension characters should be preserved as in the case of the transmission control characters and any printing characters that form later parts of escape sequences will be interpreted as stand alone characters.

d) Combined horizontal and vertcal movement of cursor positioning will not be preserved.

2. 8859/1

The Latin Alphabet No. 1 will be used to specify the printable rendition of C0 and C1.

The C0 control characters and their associated rules will be taken from the IA5 definition.

The C1 control characters will simply have their codes preserved across a transfer.

10.2.7.2 Document Type Negotiation Rules

1. Connection Establishment:

DocumentTypeNames are to be negotiated by subset (of the proposed base DocumentTypeNames) without regard to DU syntax parameter(s) that may be supplied on any DocumentTypeNames which require a DU syntax specification.

2. File Creation:

An F-CREATErequest must contain a DocumentTypeName from the negotiated set of base DocumentTypeNames. If the DocumentTypeName being used requires DU syntax parameters, then these parameters must be supplied. If the DocumentTypeName being used requires DU syntax parameters and none are provided on the F-CREATErequest, then the F-CREATErequest must be rejected.

3. File Opening:

It is recommended that the F_OPENrequest use the DocumentTypeName form (with appropriate DU syntax parameters), when proposing a Contents Type, in preference to the Constraint Set Name and Abstract Syntax Name form.

Similarly, an F_OPENresponse should use the DocumentTypeName option (with appropriate DU syntax parameters) in the Contents Type field. This will allow the receiving entity to use the DocumentTypeName attributed to the file instead of receiving a Constraint Set Name and Abstract Syntax Name pair which does not reflect the file information contained in the NBS document types.

NOTE: An F_OPENresponse without a DocumentTypeName (but carrying the Constraint Set Name and Abstract Syntax name form may cause the initiator to issue an F_CLOSErequest.

10.2.7.3 Relationship Between DUs, DEs and Document Types

"Abstract Syntax" is used to refer to the syntactic information which is architecturally passed between the Application and Presentation layers. The Abstract Syntax defines data element (DE) types. The DE types are not necessarily ASN.1 primitive types. Data types may be made up of other data types. Data Elements are not defined in terms of other data elements.

A data unit (DU) is a sequence of one or more data elements. Architecturally, entire, single DEs are passed into and out of the application process. In a real implementation, DUs may be passed.

In order to maintain DU boundaries during transfer, file structuring information must be passed (ISO8571-FADU DEFINITIONS, FTAM Part 2 section 5.3.2). A data element is referred to as a File Contents Data Element in ISO8571-FADU DEFINITIONS.

Document types refer to aspects of local processing and storage. Document types describe:

1) structural relationship between DUs,
2) structure of DUs, called DU syntax, and
3) data element types found in the file.

Because document types have to do with local processing and storage, the DU syntax makes assertions about the syntax, and size of DUs (records) in storage. Parameters on the document types provide this information about the syntax and size of the DUs.

10.2.8 F-CANCEL Action

When an F-CANCEL is sent or received, the following will occur:

- no more data will be sent
- check point numbers are removed
- state of the file will be implementation dependent.

10.2.9 Error Handling Agreements

1. Diagnostic is mandatory only when the Action Result or State Result is not zero.

2. General catch-all diagnostic is discouraged.

3. Further details subfield is mandatory. Use of octet string is discouraged.

4. Use of F-P-ABORT for other than protocol errors and catastrophic situations is discouraged.

5. When returning an error status in a file management related diagnostic (i.e., F-READ-ATTRIBUTE-response or F-CHANGE-ATTRIBUTE-response), identify the erroneous attribute by using the first two characters of Diagnostic further-details to hold a 2-digit number (encoded in IA5) from the F-READ-ATTRIBUTE-request attributes abstract syntax definition (ISO/DIS 8571/4 section 20-4):

00	Filename
01	Contents-Type
02	Storage Account
03	Date and Time of Creation
04	Date and Time of Last Modification
05	Date and Time of Last Read Access
06	Date and Time of Last Attribute Modification
07	Identity of Creator
08	Identity of Last Modifier
09	Identity of Last Reader
10	Identity of Last Attribute Modifier
11	File Availability
12	Permitted Actions
13	Filesize
14	Future Filesize
15	Access Control
16	Encryption Name
17	Legal Qualifications
18	Private Use

10.2.10 Concurrency

The concurrency control used by default for the first accessor of a file is:

read	shared
insert	exclusive
replace	exclusive
extend	exclusive
erase	exclusive
rattr	shared
cattr	exclusive
del file	exclusive

For subsequent file accessors, the requested access and processing mode service parameters are checked against this, and access given only if the request is for operations that are shared.

10.2.11 Security

1. Users may provide "user Id" (Initiator Identity) and "password" (filestore password). If the information is provided, the information will be sent to the Responder on the F-INITIALIZE.

2. Users may provide "access passwords". If the information is provided, the passwords will be sent to the Responder in the "access passwords" parameter.

3. It is the responsibility of each local system to provide security of its own real file store.

4. The syntax of "user Id" and "password" (filestore password) is system-dependent.

5. Encryption of passwords will not be done by FTAM.

6. "User Id" and "password" will represent 'account' information (this may be different from the 'account' parameter).

7. A user of the file service must be known by the Responder.

8. A commonly defined "anonymous user" convention will be provided for all systems that choose to support this capability. Access available to that user is locally determined. The ID to be used is ANON. Any password should succeed.

9. Password support in FTAM is not required.

10.2.12 Negotiation

The guidelines for negotiation in the following table have been agreed.

Table 10.2 FTAM negotiation rules

Service or Parameter	Depends on or may be negotiated down by
F-INITIALIZE	
Req.Pres.ContextMgmt	Success or failure.
Req.Func. Units	Negotiated by subset. (Affects session functional units.)
Req.Attr. Groups	Negotiated by subset.
Req.Comm.Quality of services	Reference session.
Req.ContentTypeList	Negotiated by subset.
F-SELECT	
Attributes	Only by filename. N.B. the filename on response and confirm must be that of an existing virtual file.

Requested Access	Negotiated by subset (in case of complete or partial success), must be consistent with Functional Units negotiated, access control attribute and permitted actions attribute.
F-CREATE	Consistent with Functional Units.
Initial attributes	1) The attributes returned are within the subset negotiated at initialization 2) The individual attribute values returned must be consistent with negotiation/ranges for that attribute. 3) The Responder returns values for all attributes which differ from the actual request.
Requested Access	as in F-SELECT
F-DELETE	Consistent with Functional Units.
F-READ-ATTR	Consistent with Functional Units, service class and requested access.
F-CHANGE-ATTR	as for F-READ-ATTRIBUTE
Attributes	If any attribute cannot be successfully changed, then an error more severe than warning should be returned and no attribute should be changed.
F-OPEN	
Processing Mode	Not Negotiated. Must be consistent with Functional Unit and Requested Access negotiated, with the permitted actions attribute and the contents type name.
Contents Type	Defined in DIS.
Concurrency Control	More restrictive than concurrency control of F-SELECT; consistent with the concurrency control of other users.
[F-BEGIN-GROUP F-END-GROUP]	Consistent with Functional Units and service class.
[F-LOCATE F-ERASE]	Consistent with Functional Units, requested access (and therefore also permitted actions). Service Class and Processing Mode

F-READ F-WRITE	Functional Units, Service Class and number of bulk data transfers, Requested Access and Processing Mode.
F-DATA/F-DATA-END F-CANCEL F-TRANSFER-END	Functional Units and Service Class.

10.2.13 Presentation Context Negotiation

After successful negotiation of a presentation context between two applications, a presentation context, i.e. a pair of (abstract syntax, transfer syntax), has been agreed between the application and presentation entities. The transfer syntax may have a wider range than the associated abstract syntax, i.e., it may also encode data types not contained in that abstract syntax.

The understanding (for the Implementors Agreements) is that for that specific application association the transfer syntax is restricted to the abstract syntax. That means encodings of data types not contained in the negotiated abstract syntax may be rejected by the receiving presentation entity.

10.2.13.1 Steps of Presentation Context Negotiation

There are four entities involved in the negotiation of a Presentation Context, two Application Entities, and two Presentation Entities. The relationship among the entities is as shown in the figure below:

```
Application          AE1           AE2
                      |             |
                  ----0----     ----0----
                      |             |
Presentation         PE1           PE2
                      |             |
                  ----0----     ----0----
Lower Layers          |_____________|
```

Figure 10.1 Relationship among entities

AE1 - Application Entity 1 - needs an additional abstract syntax
AE2 - Application Entity 2
PE1 - Presentation Entity on AE1's node
PE2 - Presentation Entity on AE2's node

Once AE 1 determines that it has a need to use an additional abstract syntax, there are seven steps in the negotiation of a Presentation Context. They are:

1) AE1 informs PE1 that it needs a new abstract syntax, say AS1.

2) PE1 picks a favorite transfer syntax, or set of transfer syntaxes, that can encode AS1, say TS1.

3) PE1 asks PE2 if it can deal with TS1. It does this by sending the pair (AS1,TS1) to PE2 along with an identifier for the Presentation Context.

4) PE2 decides if it can handle the transfer syntax TS1. If it can, it informs AE2 that there has been a request to use AS1.

5) AE2 decides if it can handle AS1, and informs PE2.

6) If PE2 has accepted TS1 and AE2 has accepted AS1 then PE2 informs PE1 that the Presentation Context has been accepted. If either TS1 or AS1 has been rejected, PE2 informs PE1 that the Presentation Context has been rejected.

7) PE1 tells AE1 whether AS1 has been accepted or rejected.

10.2.14 Conformance

The FTAM SIG in cooperation with the COS FTAM Technical Committee will propose a revised conformance statement for the next version of these agreements.

10.2.15 Migration Strategy

NBS Phase 2 FTAM:

There is no backward compatibility with NBS FTAM Phase 1. The following examples provide some of the technical clarifications as to why this backward compatibility is impossible:

Phase 1	Phase 2
Uses Session directly	Uses CASE services
	Uses Presentation Services
	Filestore differences
	PDU Abstract syntax differences
	FADU Structuring abstract syntax differences
	Transfer syntax differences
	Transparency mechanism differences
	Service Class Negotiation differences

NBS FTAM Future Phases:

Given that FTAM Phase 2 is based on the forthcoming FTAM IS and that this IS will provide the ability to pass "user version" information and will provide backward compatibility of protocol between versions of the IS, the workshop will specify mechanisms to provide FTAM product backward compatibility for one previous product version (i.e., NBS Phase).

11. ISO PRESENTATION LAYER PROTOCOL

Presentation Services are defined to meet the needs of many applications. The Presentation service is defined in ISO DIS 8822 and the Presentation protocol is defined in ISO DIS 8823. The general agreements about the implementation of Presentation are documented below.

11.1 General

1) Implementations will be based on the DIS (to be replaced by the IS) version of the ISO Presentation service and protocol.

2) A conformant implementation must be conformant to the ISO Presentation service and protocol, and must meet all of the requirements of this specification.

11.2 Functional Units

1) The following functional units are mandatory for implementation.

   - Kernel

2) Implementation of any other functional unit is optional.

11.3 Abstract Syntaxes

1) The following abstract syntaxes must be supported.

- ISO 8571-FTAM	(FTAM PCI)
- ISO 8571-FADU	(File Structuring)
- ISO 8650-2-ACSE1	(CASE PCI)
- NBS-AS1	(Primitive data types)

2) The following abstract syntaxes are defined in these agreements, but are optional.

- NBS-AS2	(floating point numbers)
- NBS-AS3	(file directories)

3) Any other abstract syntax may be supported.

4) Abstract syntaxes will be identified by registered name.

11.4 Transfer Syntaxes

1) The following transfer syntaxes must be supported for all mandatory abstract syntaxes (also for NBS-AS2 and NBS-AS3 if implemented).

   - NBS-TS1

2) Any other transfer syntax may be implemented.

3) Transfer syntaxes will be identified by registered name.

12. COMMON APPLICATION SERVICE ELEMENTS PROTOCOL

Presently, the following agreements apply only when using FTAM.

CASE Part 2 Association Control Services are defined to meet the needs of many applications. The CASE Part 2 Association Control Service is defined in ISO DIS 8649/2 and the CASE Part 2 Association Control Protocol is defined in ISO DIS 8650/2. The general agreements about the implementation of CASE are documented below.

12.1 General

1) Implementations will be based on the DIS version of ISO 8649/2 and ISO 8650/2.

2) A conformant implementation must be ISO conformant as well as meet all of the requirements of this specification.

3) All services specified in ISO DIS 8649/2 must be implemented.

12.2 Application Contexts

1) The following application contexts must be implemented.

   - ISO FTAM

12.3 Application Entity Titles

1) Application Entity Title will name a specific application entity on a specific node (AE Instance).

2) An AE Instance corresponds to exactly one application process.

3) The naming of application entities is external to this specificaion.

4) Application entity titles may have a many-to-one mapping to a PSAP address (i.e., aliases may be used).

5) An application entity title is a registered name of type OBJECT IDENTIFIER.

6) The application entity invocation does not have an address.

13. X.400 BASED MESSAGE HANDLING SYSTEM

13.1 INTRODUCTION

This is an implementation agreement developed by the Implementor's Workshop sponsored by the U.S. National Bureau of Standards to promote the useful exchange of data between devices manufactured by different vendors. This agreement is based on, and employs protocols developed in accord with, the OSI Reference Model. While this agreement introduces no new protocols, it eliminates ambiguities in interpretations.

This is an implementation agreement for a Message Handling System (MHS) based on the X.400-series of Recommendations (1984) from the CCITT. Figure 13.1.1 displays the layered structure of this agreement.

This agreement can be used over any transport profile. In particular, this profile can be used over the transport protocol class 0 used over CCITT X.25, described in Section 7.2 of this document. In addition, this profile can be used over the transport profiles used in support of MAP (Manufacturing Automation Protocols) or TOP (Technical and Office Protocols). Note that the MAP or TOP environment must support the reduced Basic Activity Subset (BAS) as defined in X.410.

The UAs and MTAs require access to directory and routing services. A Directory Service is to be provided for each (vendor-specific) domain. Except insofar as they must be capable of providing addressing and routing described hereunder, these services and associated protocols are not described by this agreement.

The material on PRMD-PRMD message transfer in this implementation specification is intended to be stable enough to provide a reliable guide to implementation of X.400. The material on ADMD-PRMD and ADMD-ADMD is incomplete and serves as an indication of direction.

User Agent Layer	CCITT X.420
Message Transfer Agent Layer	CCITT X.411
Reliable Transfer Service Layer	CCITT X.410
Presentation Layer	CCITT X.410 sec. 4.2
Session Layer	CCITT X.225

Figure 13.1.1 The layered structure of this implementation agreement

13.2 SCOPE

This agreement applies to Private Management Domains (PRMDs) and Administration Management Domains (ADMDs). Three boundary interfaces are specified:

A) PRMD to PRMD;
B) PRMD to ADMD;
C) ADMD to ADMD.

In case (A), the PRMDs do not make use of MHS services provided by an ADMD. In cases (B) and (C), UAs associated with an ADMD can be the source or destination for messages. Furthermore, in cases (B) and (C), an ADMD can serve as a relay between MDs. Figure 13.2.1 illustrates the interfaces to which the agreement applies.

X.400 protocols other than the Message Transfer Protocol (P1) and the Interpersonal Messaging Protocol (P2) are beyond the scope of this agreement. Issues arising from the use of other protocols or relating to P1 components in support of other protocols are outside the scope of this document. This agreement describes the minimum level of services provided by Management Domains (MDs). Provision for the use of the remaining services defined in the X.400 Series of Recommendations is outside the scope of this document.

This agreement does not cover message exchange between communicating entities within a domain even if these entities communicate via P1 or P2. Bilateral agreements between domains may be implemented in addition to the requirements stated in this document. Conformance to this agreement requires the ability to exchange messages with conforming domains that have made no bilateral agreements.

PRMD = Private Management Domain

ADMD = Administration Management Domain

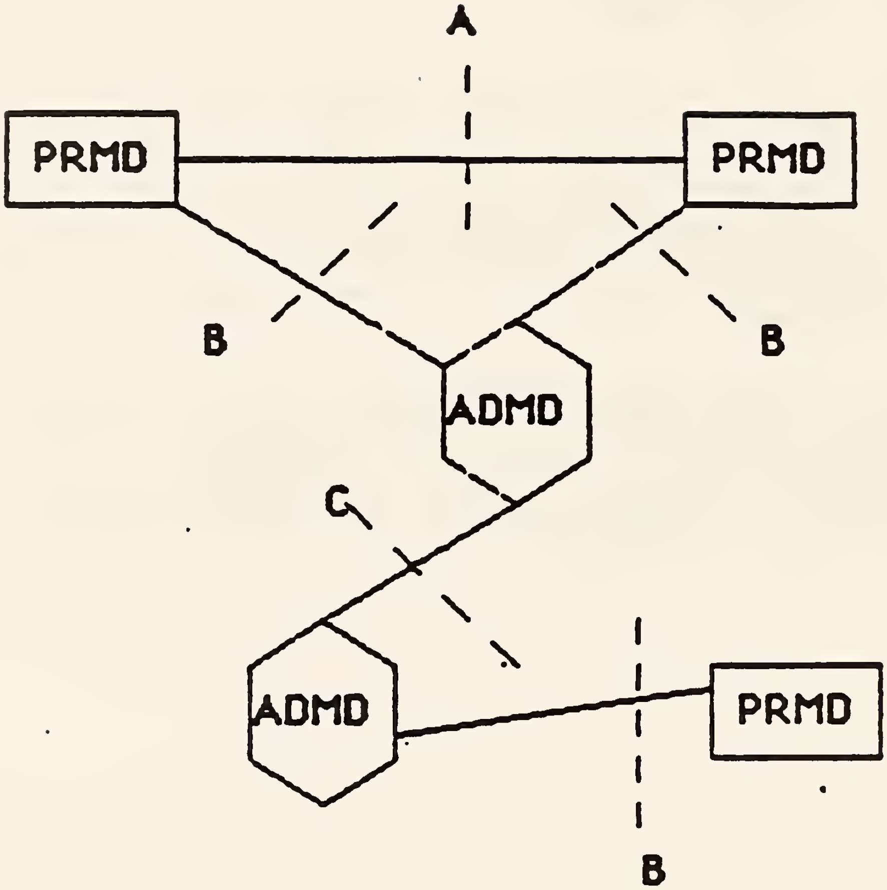


Figure 13.2.1 This agreement applies to the interface between: (A) PRMD and PRMD; (B) PRMD and ADMD; (C) ADMD and ADMD

13.3 PRMD to PRMD

This section is limited in scope to issues arising from the direct connection (interface A in Figure 13.2.1) of two PRMDs. "Direct" means that no ADMD provides MHS services to facilitate message interchange. "Direct" does not exclude those instances for which ADMDs provide lower layer services (e.g., X.25). Figure 13.3.1 schematically represents the scope of this section.

These issues relate to the use of the UAL (User Agent Layer) and MTL (Message Transfer Layer) services, protocol elements, recommended practices and constraints. In particular, this section addresses the P1 and P2 protocols and their related services in a direct connection environment. This section describes the minimum level of services provided by a PRMD. Provision for the use of the remaining services defined in the X.400 Series of Recommendations is beyond the scope of this section.

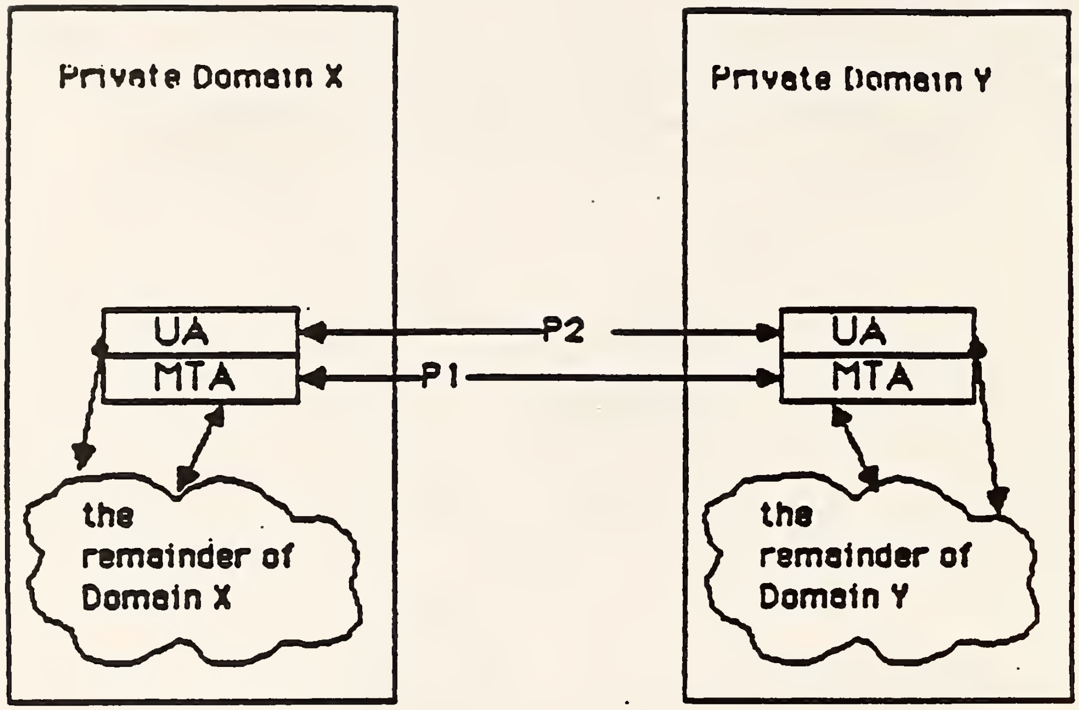


Figure 13.3.1 Interconnection of private domains

13.3.1 Service Elements and Optional User Facilities

This section identifies those service elements and optional user facilities that must be provided in support of P1 and P2.

13.3.1.1 Classification of Support for Services

The classification of UA and MT-Service elements is used to define characteristics of equipment. Equipment can claim SUPPORT or NON-SUPPORT of a Service; in the case of UA-service elements, a separate classification is given for Origination and Reception.

The service provider is defined as the entity providing the service, in this case, the MTL or the UAL. The service user is either the MHS user or the UAL. The classification of provider and user relates to the sublayer for which the service element is defined.

13.3.1.1.1 Support (S)

This means that:

a. The service provider makes the service element available to the service user.

b. The service user gives adequate support to the MHS to invoke the service element or makes information associated with the service element available.

Support for Origination means that:

a. The service provider makes the service element available to the service user for invocation.

b. The service user gives adequate support to the end user of the MHS to invoke the service element.

Support for Reception means that the service provider makes information associated with the service element available to the service user.

Note: A UA- or MT-service element can carry information from originator to recipient only if:

° the service element is available to the originator,
° the service element is available to the recipient, and
° all intermediate steps carry the information.

13.3.1.1.2 Non-Support (N)

This means that the service provider is not required to make the service element available to the service user. However, the service provider should not regard the occurrence of the corresponding protocol elements as an error and should be able to relay such elements. Implementations making a profile available should indicate deviations (additions or deletions) with respect to the requirement in the profile.

13.3.1.1.3 Not Used (N/U)

This means that although the Recommendations allow this service element, this profile does not use it.

13.3.1.1.4 Not Applicable (N/A)

This means that this service element does not apply in this particular case (for originator or recipient).

13.3.1.2 Summary of Supported Services

a. Within a PRMD, a User Agent must support all P2 BASIC IPM Services (X.400) and all P2 ESSENTIAL IPM Optional user facilities (X.401) subject to the qualifiers listed in APPENDIX A.

b. Within a PRMD, a MTA must support all BASIC MT Services (X.400) and all ESSENTIAL MT optional user facilities (X.401) subject to the qualifiers listed in APPENDIX A.

c. No support is required of the additional optional user facilities of X.401.

13.3.1.3 MT Service Elements and Optional User Facilities

Tables 13.3.1 through 13.3.3 show the message transfer (MT) service elements and optional user facilities.

Table 13.3.1 Basic MT service elements

Service Elements	Support (S) or Non-support (N)
Access Management	N/U (a)
Content Type Indication	S
Converted Indication	S
Delivery Time Stamp Indication	S
Message Identification	S
Non-delivery Notification	S
Original Encoded Information Types Indication	S
Registered Encoded Information Types	N/U (a)
Submission Time Stamp Indication	S

a: Not applicable to co-resident UA and MTA.

Table 13.3.2 MT optional user facilities
provided to the UA-selectable on a per-message basis

MT Optional User Facilities	Categorization	Support (S) or Non-support (N)
Alternate Recipient Allowed	E	S
Conversion Prohibition	E	S
Deferred Delivery	E	N (b)
Deferred Delivery Cancellation	E	N (b)
Delivery Notification	E	S
Disclosure of Other Recipients	E	N (c)
Explicit Conversion	A	N
Grade of Delivery Selection	E	S
Multi-destination Delivery	E	S
Prevention of Non-delivery Notification	A	N
Probe	E	N (d)
Return of Contents	A	N

Table 13.3.3 MT optional user facilities
provided to the UA agreed for a contractual period of time

MT Optional User Facilities	Categorization	Support (S) or Non-Support (N)
Alternate Recipient Assignment	A	N
Hold for Delivery	A	N/U
Implicit Conversion	A	N

E: Essential optional user facility.
A: Additional optional user facility.
b: A local facility subject to qualifiers in APPENDIX A.
c: Support not required for an originating MT user; support must be provided for recipient MT users.
d: Subject to qualifiers in APPENDIX A.

13.3.1.4 IPM Service Elements and Optional User Facilities

Tables 13.3.4 through 13.3.6 show the IPM service elements and optional user facilities.

Table 13.3.4 Basic IPM service elements

Service Elements	Origination by UAs	Reception by UAs
Access Management	N/U (a)	N/U (a)
Content Type Indication	S	S
Converted Indication	N/A	S
Delivery Time Stamp Indication	N/A	S
Message Identification	S	S
Non-delivery Notification	S	N/A
Original Encoded Information Types Indication	S	S
Registered Encoded Information Types	N/A	N/A (a)
Submission Time Stamp Indication	S	S
IP-message Identification	S	S
Typed Body	S	S

(a) Does not apply to co-resident UA and MTA.

Table 13.3.5 IPM optional facilities agreed for a contractual period of time

Service Elements	Categorization	Support (S) or Non-Support (N)
Alternate Recipient Assignment	A	N
Hold for Delivery	A	N
Implicit Conversion	A	N

Table 13.3.6 IPM optional user facilities selectable on a per-message basis

IPM Optional User Facilities	Origination by UAs	Reception by UAs
Alternate Recipient Allowed	A (N)	A (N)
Authorizing Users Indication	A (N)	E (S)
Auto-forwarded Indication	A (N)	E (S)
Blind Copy Recipient Indication	A (N)	E (S)
Body Part Encryption Indication	A (N)	E (S)
Conversion Prohibition	E (S)	E (S)
Cross-referencing Indication	A (N)	E (S)
Deferred Delivery	E (N) (f)	N/A
Deferred Delivery Cancellation	A (N/U) (f)	N/A
Delivery Notification	E (S)	N/A
Disclosure of Other Recipients	A (N)	E (S)
Expiry Date Indication	A (N)	E (S)
Explicit Conversion	A (N)	N/A
Forwarded IP-message Indication	A (N)	E (S)
Grade of Delivery Selection	E (S)	E (S)
Importance Indication	A (N)	E (S)
Multi-destination Delivery	E (S)	N/A
Multi-part Body	A (N)	E (S)
Non-receipt Notification	A (N)	A (N)
Obsoleting Indication	A (N)	E (S)
Originator Indication	E (S)	E (S)
Prevention of Non-delivery Notification	A (N)	N/A
Primary and Copy Recipients Indication	E (S)	E (S)
Probe	A (N)	N/A
Receipt Notification	A (N)	A (N)
Reply Request Indication	A (N)	E (S)
Replying IP-message Indication	E (S)	E (S)
Return of Contents	A (N)	N/A
Sensitivity Indication	A (N)	E (S)
Subject Indication	E (S)	E (S)

f: A local facility subject to qualifiers in APPENDIX A.

13.3.2 X.400 Protocol Definitions

13.3.2.1 Introduction

This section reflects the agreements of the NBS/OSI Workshop regarding P1 and P2 protocol elements.

13.3.2.1.1 Protocol Classification

The protocol classifications are defined:

a) UNSUPPORTED = X

These elements may be generated, but no specific processing should be expected in a relaying or delivering domain. A relaying domain must at least relay the semantics of the element. The absence of these elements should not be assumed, in a relaying or delivering domain, to convey any significance.

b) SUPPORTED = H

These elements may be generated. However, implementations are not required to be able to generate these elements. Appropriate actions shall be taken in a relaying or delivering domain.

c) GENERATABLE = G

Implementations must be able to generate and handle these protocol elements, although they are not necessarily present in all messages generated by implementations of this profile. Appropriate actions shall be taken in a relaying or delivering domain.

d) REQUIRED = R

Implementations of this profile must always generate this protocol element. However, its absence cannot be regarded as a protocol violation as other MHS implementations may not require this protocol element. Appropriate actions shall be taken in a relaying or delivering domain.

e) MANDATORY = M

This must occur in each message as per X.411 or X.420 as appropriate; absence is a protocol violation. Appropriate actions shall be taken in a relaying or delivering domain.

13.3.2.1.2 General Statements on Pragmatic Constraints

a. Where a protocol element is defined as a choice of Numeric String and Printable String (i.e., Country Name, Administration Domain Name and Private Domain Identifier), then a numeric value encoded as a printable string is equivalent to the same value encoded as a numeric string.

b. The maximum number of recipients in a single MPDU is 32K - 1 (that is, 32767). However, no individual limits on the number of occurrences (recipients) are placed on the following protocol elements: Authorizing Users, Primary Recipients, Copy Recipients, Blind Copy Recipients, Obsoletes and Cross References. Additionally, there is no limit on the number of Reply to Users. This is a local matter for the originating system.

c. Use of strings. A Printable String is defined in terms of the number of characters, which is the same number of octets. For T.61 strings the number of octets is twice the number of characters specified.

d. The ability to generate maximum size elements is not required, with the exception of the component fields in the Standard Attribute List, in which case it is required.

13.3.2.1.3 MPDU Size

The following agreements govern the size of MPDUs:

a. All MTAEs must support at least one MPDU of at least one megabyte.

b. The size of the largest MPDU supported by a UAE is a local matter.

13.3.2.2 P1 Protocol Elements

13.3.2.2.1 P1 Envelope Protocol Elements

Table 13.3.7 contains Protocol Elements and their classes.

Table 13.3.7 P1 protocol elements

Element	Class	Restrictions and Comments
MPDU		
UserMPDU	G	
DeliveryReportMPDU	G	
ProbeMPDU	H	
UserMDPU		
UMPDUEnvelope	M	
UMPDUContent	M	
UMPDUEnvelope		
MPDUIdentifier	M	
originator	M	
originalEncodedInformationTypes	G	If this field is absent, then the Encoded Information Type is "unspecified".
ContentType	M	
UAContentID	H	Maximum length = 16 characters.
Priority	G	
PerMessageFlag	G	Maximum length = 2 octets.
deferredDelivery	X	
PerDomainBilateralInfo	X	No limit on number of occurrences
RecipientInfo	M	Maximum number = 32K - 1 occurrences. More severe limitations are by bilateral agreement.
TraceInformation	M	
UMPDUContent	M	
MPDUIdentifier		
GlobalDomainIdentifier	M	
IA5String	M	Maximum length = 32 characters, graphical subset only. Refer to T.50 for clarification of graphical subset.
PerMessageFlag		
discloseRecipients	H	
conversionProhibited	G	
alternateRecipientAllowed	H	
contentReturnRequest	X	
PerDomainBilateralInfo		
CountryName	M	Maximum length = 3 characters.
AdministrationDomainName	M	Maximum length = 16 characters.
BilateralInfo	M	Maximum depth = 8; maximum length = 1024 octets (including encoding).

(continued on next page)

Table 13.3.7 P1 protocol elements, Continued

Element	Class	Restrictions and Comments
RecipientInfo		
recipient	M	
ExtensionIdentifier	M	Maximum value = 32K - 1 (32767).
perRecipientFlag	M	Maximum length = 2 octets.
ExplicitConversion	X	
perRecipientFlag		
ResponsibilityFlag	M	
ReportRequest	M	
UserReportRequest	M	
TraceInformation		Reference should be made to Version 3 of the X.400 Implementor's Guide for information related to Trace sequencing.
GlobalDomainIdentifier	M	
DomainSuppliedInfo	M	
DomainSuppliedInfo		
arrival	M	
deferred	X	
action	M	
converted	H	
previous	X	
ORName		
EncodedInformationTypes		
bit string	M	Delivery can only occur if match is made with Registered Encoded Information Types. Individual vendors may impose limits. Maximum length = 3 octets.
G3NonBasicParameters	X	
TeletexNonBasicParameters	X	
PresentationCapabilities	X	
DeliveryReportMPDU		
DeliveryReportEnvelope	M	
DeliveryReportContent	M	
DeliveryReportEnvelope		
report	M	
originator	M	
TraceInformation	M	
DeliveryReportContent		
original	M	
intermediate	X	
UAContentID	G	
ReportedRecipientInfo	M	Maximum number = 32K - 1 occurrences.
returned	H	Can only be issued if specifically requested in the originating message.

(Continued on next page)

Table 13.3.7 P1 protocol elements, continued

Element	Class	Restriction and Comments
billingInformation	X	Maximum depth = 8; maximum length = 1024 octets (including encoding).
ReportedRecipientInfo		
recipient	M	
ExtensionsIdentifier	M	
PerRecipientFlag	M	
LastTraceInformation	M	
intendedRecipient	H	
SupplementaryInformation	X	Maximum length = 64 characters. <u>NOTE</u>: This is subject to change. Value is pending verification by the CCITT SG VIII or IX.
LastTraceInformation		
arrival	M	
converted	H	
Report	M	
Report		
DeliveredInfo	G	Generated if delivery is reported
NonDeliveredInfo	G	Generated if failure to deliver is reported.
DeliveredInfo		
delivery	M	
typeofUA	R	This element must be generated by PRMDs with a PRIVATE value.
NonDeliveredInfo		
ReasonCode	M	
DiagnosticCode	H	Whenever possible, use a meaningful diagnostic code.
ProbeEnvelope		
probe	M	
originator	M	
ContentType	M	
UAContentID	H	Maximum length = 16 characters.
original	G	If this field is absent, then the Encoded Information Type is "unspecified".

(Continued on next page)

Table 13.3.7 P1 protocol elements, continued

TraceInformation (under Probe Envelope)	M	
PerMessageFlag	G	
contentLength	H	
PerDomainBilateralInfo	X	
RecipientInfo	M	Maximum number = 32K - 1 occurrences.
	--End of Definitions	

13.3.2.2.2 ORName Protocol Elements

Only form 1 variant 1 O/R names are supported.

Table 13.3.8 contains ORName Protocol Elements.

Table 13.3.8 ORName protocol elements

Element	Class	Restrictions and Comments
ORName		
StandardAttributeList	M	
DomainDefinedAttributeList	X	
StandardAttributeList (1)		
CountryName	R	As defined in X.411, Maximum length = 3 characters.
AdministrationDomainName (4)	R	Maximum length = 16 characters or digits.
X.121Address	X	Maximum length = 15 digits.
TerminalID	X	Maximum length = 24 characters.
PrivateDomainName (2)	G	Maximum length = 16 characters.
OrganizationName (2)	G	Maximum length = 64 characters.
UniqueUAIdentifier	X	Maximum length = 32 digits.
PersonalName	G	Maximum length of values of sub-elements = 64 characters. NOTE: The possibility that this value may be reduced to 40 characters is for further study by the CCITT.
OrganizationalUnit (3)	G	Maximum length = 32 characters per occurrence. A maximum of four occurrences are allowed.
DomainDefinedAttributeList (5)	X	Maximum = 4 occurrences.
type	M	Maximum length = 8 characters.
value	M	Maximum length = 128 characters.
PersonalName		
surName	M	Maximum length = 40 characters.
givenName	G	Maximum length = 16 characters.
initials	G	Maximum length = 5 characters; excluding surname initial and punctuation and spaces.
generationQualifier	G	Maximum length = 3 characters.

GlobalDomainIdentifier		
CountryName	M	Maximum length = 3 characters.
AdministrationDomainName (4)	M	Maximum length = 16 characters or digits.
PrivateDomainIdentifier	R	Maximum length = 16 characters or digits. This element must be generated by PRMDs.

--End of Definitions--

(Continued on next page)

Table 13.3.8 ORName Protocol Elements, Continued

Notes:

(1) The following apply for comparison of the Standard Attributes of an O/R Name:

1) Lower case is interpreted as upper case (for IA5).
2) Multiple spaces may be interpreted as a single space. Originating domains shall only transmit single significant spaces. If multiple spaces are transmitted, non-delivery may occur.

(2) At least one of these must be supplied.

(3) These should be sent in ascending sequence, from the least significant <Organizational Unit> (lowest in organization hierarchy) to the most significant. Only those specified should be sent. (That is, an unspecified <Organizational Unit> should not be sent along as a field of [null] content, nor zero length, etc.)

(4) This attribute shall contain one space in all ORNames of messages originated in a PRMD that is not connected to an ADMD, and in ORNames of recipients reachable only through a PRMD; otherwise, this attribute shall contain an appropriate ADMD name.

(5) Many existing mail systems require attributes not present in these agreements. Domain Defined Attributes are a method of providing these. Failure to support the specification of DDAs may prevent successful interworking with such existing mail systems until such time as all mail systems are capable of supporting delivery via the standard attribute list only. Specific recommendations on the use of DDAs are in the Recommended Practices section.

13.3.2.3 P2 Protocol Profile (Based on [X.420])

Tables 13.3.9 and 13.3.10 classify the support for the P2 protocol elements required by this profile. The tables give restrictions and comment in addition to [X.420].

Restriction on length is one of the types of restrictions. The reaction of implementation to a violation of this restriction is not defined by this profile.

13.3.2.3.1 P2 Protocol - Heading

Table 13.3.9 below specifies the support for protocol elements in P2 Headings.

Table 13.3.9 P2 heading protocol elements

Element	Class	Restrictions and Comments
UAPDU		
IM-UAPDU	G	
SR-UAPDU	X	
IM-UAPDU		
Heading	M	
Body	M	
Heading		
IPMessageId	M	
originator	R	
authorizingUsers	H	
primaryRecipients	G	At least one of primaryRecipients, copyRecipients, or blindCopyRecipients must be present.
copyRecipients	G	
blindCopyRecipients	H	
inReplyTo	G	
obsoletes	H	
crossReferences	H	
subject	G	Maximum length = 256 octets; the ability to generate the maximum size subject is not required.
expiryDate	H	
replyBy	H	
replyToUsers	H	
importance	H	Appropriate action is for further study
sensitivity	H	Appropriate action is for further study
autoforwarded	H	
IPmessageId		
ORName	H	
PrintableString	M	Maximum length = 64 characters.
ORDescriptor		
ORName	H	Specify the ORName whenever it is possible. See APPENDIX B.
freeformName	H	Maximum length = 64 characters, graphical subset only (128 octets).
telephoneNumber	X	Maximum length = 32 characters. This allows for punctuation. It does not take into account possible future use by ISDN.
Recipient	M	
ORDescriptor	M	
reportRequest	X	
replyRequest	H	

(Continued on next page)

Table 13.3.9 P2 heading protocol elements, continued

Element	Class	Restrictions and Comments
Body		No limit on number of BodyParts.
BodyPart	G	No limit on length of any BodyPart or the depth of ForwardedIPMessage BodyParts nested. Classification is subject to pending CCITT resolution.
SR-UAPDU		
nonReceipt	H	
receipt	H	
reported	M	
actualRecipient	R	
intendedRecipient	H	
converted	X	
NonReceiptInformation		
reason	M	
nonReceiptQualifier	H	
comments	H	Maximum length = 256 characters.
returned	H	May only be issued if specifically requested by originator.
ReceiptInformation		
receipt	M	
typeOfReceipt	H	
SupplementaryInformation	X	Maximum length = 64 characters. <u>NOTE</u>: This value is pending verification by the CCITT SG VIII or IX.

13.3.2.3.2 P2 Protocol - BodyParts

All BodyParts with identifiers in the range 0 up to and including 16K -1 are legal and should be relayed. BodyPart identifiers corresponding to X.121 Country Codes should be interpreted as described in section 13.4.3.2.1.

13.3.2.3.2.1 Privately Defined BodyParts

This section describes an interim means for identifying privately defined BodyParts. This subsection shall be replaced in a future edition taking into account CCITT recommendations with equivalent functionality.

```
BodyPart  ::=  CHOICE
   [0]IMPLICIT IA5Text,
   [1]IMPLICIT TLX,
         .
         .
         .
   [234]IMPLICIT UKBodyParts,
         .
         .
         .
   [310]IMPLICIT USABodyParts,
         .
         .
         .                    ]
```

Where UKBodyParts and USABodyParts are defined as:

```
  SEQUENCE  BodyPartNumber, ANY

  BodyPartNumber ::= INTEGER
```

In the EncodedInformationTypes of the P1 Envelope, the undefined bit must be set when a message contains a privately defined BodyPart. Each UA that expects such BodyParts should include undefined in the set of deliverable EncodedInformationTypes it registers with the MTA.

All BodyPartNumbers assigned must be interpreted relative to the BodyPart in which it is used, which is that tagged with the value [310] in the case of those defined within the United States. The NBS assigns unique message BodyPartNumbers for privately defined formats within the United States.

Implementations are required to generate and image IA5Text.

Implementations should specify the other BodyPart types supported.

If an implementation supports a particular BodyPart type for reception, it should also be able to support that BodyPart type for reception if this is part of a ForwardedIPMessage.

For the BodyPart types currently considered, support for the protocol elements is as indicated in Table 13.3.10.

13.3.2.3.2.2 P2 BodyPart Protocol Elements

Table 13.3.10 P2 BodyParts

Elements	Class	Restrictions and Comment
BodyPart		
IA5Text	G	
TLX	X	
Voice	X	
G3Fax	X	
TIF0	X	
TTX	X	
Videotex	X	
NationallyDefined	X	
Encrypted	X	
ForwardedIPMessage	H	
SFD	X	
TIF1	X	
IA5Text		
repertoire	H	
IA5String	M	For rendition of IA5Text see APPENDIX C.
TLX		For further study by CCITT.
Voice		
Set		For further study by CCITT.
BitString	M	
G3Fax		
numberOfPages	X	
G3NonBasicParameters	X	
SEQUENCE (OF BIT STRING)	M	
BIT STRING	H	See NOTE:
G3NonBasicParameters		Support for individual elements is for further study.
TIF0		
T.73Document	M	
T.73ProtocolElement	H	See NOTE:
TTX		
numberOfPages	X	
telexCompatible	X	
TeletexNonBasicParams	X	
SEQUENCE (of T61String)	M	
T61String	H	See NOTE;
TeletexNonBasicParams		
graphicCharacterSets	X	
controlCharacterSets	X	
pageFormats	X	
miscTerminalCapabilities	X	
privateUse	X	

(Continued on next page)

Table 13.3.10 P2 BodyParts, Continued

Element	Class	Restrictions and Comments
Videotex		
SET		For further study by CCITT.
VideotexString	M	
NationallyDefined		
ANY	M	
Encrypted		
SET		For further study by CCITT.
BIT STRING	M	
ForwardedIPMessage		
delivery	H	
DeliveryInformation	H	
IM-UAPDU	M	
DeliveryInformation		
ContentType	M	
originator	M	
original	M	
Priority	G	
DeliveryFlags	M	
otherRecipients	H	
thisRecipient	M	
intendedRecipient	H	
converted	X	
submission	M	
SFD		
SFD.Document	M	
TIF1		
T.73 Document	M	

Note: This element is not an addition to the definition of the BodyPart. It is described here to show that the SEQUENCE may contain zero elements. A Problem Report has been submitted to the CCITT to clarify whether this is permissible. The NBS/OSI Workshop will adopt the CCITT decision.

13.3.3 Reliable Transfer Server (RTS)

13.3.3.1 Implementation Strategy

Based on X.410 clause 3 and X.411 clause 3.5.

13.3.3.2 RTS option selection

a) The maximum number of simultaneous associations is not limited in this profile; if the capacity of a system is exceeded, it should not initiate or accept additional associations.

b) Associations are established by the MTA which has messages to transfer.

c) Associations are released when they are not needed. Associations may also be ended prematurely due to internal problems of the RTS.

d) For both monologue and two way alternate associations, the initiator keeps the initial turn.

When establishing an RTS association, the following rules apply to the use of parameters in addition to those in X.410 clause 3.2.1:

Dialogue mode:	Monologue must be supported for this profile; two-way alternate is used only if both partners agree.
Initial turn:	Kept by the initiator of the association.

The 'priority-mechanism' and the 'transfer-time limit' are regarded as local matters.

13.3.3.3 RTS Protocol Options and Clarifications

Realization of the RTS protocol is subject to the following rules in addition to those specified in X.410 clause 4:

a. One RTS association corresponds with one or more consecutive session connections (not concurrent ones). The first is opened with ConnectionData of type OPEN, and subsequent ones are opened with type RECOVER.

b. Recovery of Session connection is only by RTS initiator.

c. Checkpoint size:

   - Checkpointing and No Checkpointing should be supported. Whenever possible, checkpointing should be used.

   - The minimum checkpointSize is 1 (that is, 1024 octets).

d. Window size:

   - Minimal value of 1 (if checkpointing is supported).

   - WindowSize = 1 means: After an S-SYNCH-MINOR request is sent, wait until the confirmation is received before issuing an S-DATA, S-SYNCH-MINOR, or S-ACTIVITY-END request.

e. APDUs should not be blocked into one activity.

f. Only one SSDU shall be transferred:

   - Between two adjacent minor synch points.

   - Between minor synch points and adjacent S-ACTIVITY-START and S-ACTIVITY-END requests.

   - Between S-ACTIVITY-START and S-ACTIVITY-END without checkpoints.

g. A monologue association is defined as follows:

   - The RTS user responsible for establishing the association is called the initiator.

   - The initiator keeps the initial turn.

   - APDUs are transferred in the direction of the initiator to the recipient only.

   - There shall be no token passing.

   - Only the initiator can effect an orderly release of the association.

h. A two-way alternate session is as described in X.410.

i. In the UserData parameter of the S-U-ABORT, the ReflectedParameter will not be used in the AbortInformation element.

j. When the S-ACTIVITY-RESUME is used to resume an activity in the same session connection as the one in which it started, this must happen immediately after the activity has been interrupted (i.e., no intervening activity can occur). Otherwise, [X.410 clause 4.3 paragraph 1] may be violated.

k. When S-ACTIVITY-RESUME is used to resume an activity started in another session connection, the following conditions must be met:

° The current session connection is of type "recover".

° The value of OldSessionConnectionIdentifier in S-ACTIVITY-RESUME must match the value of the SessionConnectionIdentifier parameter used in the S-CONNECT of the prior session connection. This value is also identical to the SessionConnectionIdentifier in the ConnectionData (in PConnect, in SS-UserData) for the current session connection.

° This must occur as the first activity of the next session connection for the same RTS-association. It must be the first, otherwise [X.410 clause 4.5.1 point 1] is violated.

Note: It is in the same RTS-ASSOCIATION because the use of S-ACTIVITY-RESUME only makes sense within the scope of one RTS association.

l. If the transfer of an APDU is interrupted before the confirmation of the first checkpoint, the value of the SynchronizationPointSerialNumber in S-ACTIVITY-RESUME should be zero.

The S-ACTIVITY-RESUME must be immediately followed by an S-ACTIVITY-DISCARD.

m. In S-TOKEN-PLEASE, the UserData parameter shall contain an integer conforming to X.409 which conveys the priority.

n. The receiving RTS can use the value of the Reason parameter in the S-U-EXCEPTION-REPORT to suggest to the sending RTS that it should either interrupt or discard the current activity.

As stated in Version 3 of the X.400 Series Implementor's Guide, "On receipt of an 'unrecoverable procedure error' the current activity is not recoverable and the sending RTS issues an S-ACTIVITY-DISCARD. On receipt of any other reason code (including an undefined value), the sending RTS issues an S-ACTIVITY-INTERRUPT followed by an S-ACTIVITY-RESUME."

o. In the case of S-P-ABORT, the current activity (if any) is regarded as interrupted, rather than discarded.

p. The following table illustrates the legal negotiation possibilities allowed by X.410 clause 4.2.1 regarding checkpoint size and window size:

Table 13.3.11 Checkpoint window size of IP

		acceptor answer		

		CS = 0
		(or unspecified)
		WS unspecified
			CS = m
			WS = j
			(or unspecified)
				CS = n
				WS = j
				(or unspecified)

initiator proposal	CS = 0
	(or unspecified)
	WS = i
	(or unspecified)
		legal	legal	legal

	CS = k
	WS = i
	(or unspecified)
		legal	legal	not allowed

Legend:

- CS means CheckpointSize
- WS means WindowSize
- i, j, k, m, and n are integer values with the following relations:

  $0 < m < k < n$ (values assigned to CS)
  $0 < j < i$ (values assigned to WS)

- For unspecified parameters, the default applies. In this case, the numeric relations apply, that is, the default values substitute for the unspecified integer.

13.3.3.4 RTS Protocol Limitations

The RTS Protocol Limitations for this profile are listed in Table 13.3.12.

Table 13.3.12 RTS protocol elements

Element	Class	Restriction
PConnect	M	
DataTransferSyntax	M	Value = 0.
pUserData	M	
checkpointSize	H	
windowSize	H	
dialogueMode	H	
ConnectData	M	
applicationProtocol	R	Value = 1.
ConnectionData		
open	G	
recover	G	
open		
RTS User Data	G	
recover		
SessionConnectionID	G	
RTS User Data		
mTAName	G	Maximum length 32 characters graphic subset of IA5 only.
password	G	Maximum length 64 octets graphic subset of IA5 only.
< null RTS User Data >	G	Generated if other validation methods are used.
SessionConnectionIdentifier		
CallingSSUserReference	M	Maximum length 64 octets including encoding = 62 octets of T.61.
CommonReference	M	
AdditionalReference Information	G	Maximum length 4 octets including encoding = 2 octets of T.61.
PAccept	G	
DataTransferSyntax	M	Value = 0.

(Continued on next page)

Table 13.3.12 RTS protocol elements, continued

Element	Class	Restriction
pUserData	M	
checkpointSize	H	
windowSize	H	
ConnectionData	G	
PRefuse	G	
RefuseReason	M	
SS User Data (in S-TOKEN-PLEASE)	G	
AbortInformation (in S-U-ABORT)	G	
AbortReason	H	
reflectedParameter	X	Restricted to 8 bits.

13.3.4 Use of Session Services

The session requirements and use of session are covered in section 8 of this document.

13.3.5 Data Transfer Syntax

This section defines Presentation Transfer Syntax and notation rules applicable to these agreements. Implementations must conform EXACTLY as specified in X.409 with no further restrictions. APPENDIX C defines rendition of IA5 Text and T61 characters.

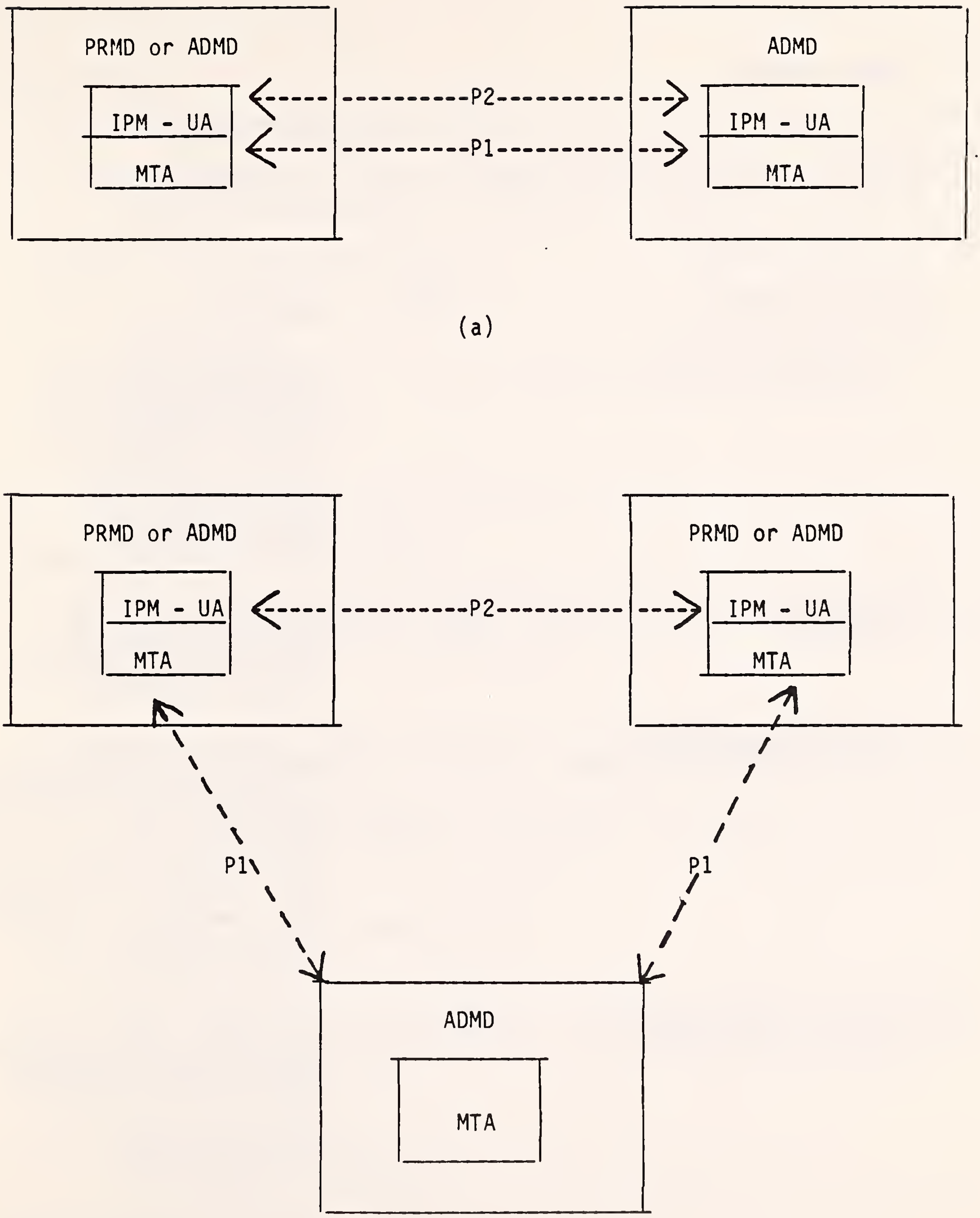
13.4 PRMD to ADMD and ADMD to ADMD

13.4.1 Introduction

This section defines the implementation agreements that apply to the interface between two management domains when at least one is an ADMD. A message arriving at an ADMD has either no recipient within that domain or one or more recipients within that domain. In the former case, the ADMD serves as a relay between two or more domains and the actions required of that ADMD are independent of the nature (PRMD or ADMD) of the domains. In the latter case, the ADMD is responsible for delivering messages to the proper recipient(s) within its jurisdiction, and may also be responsible for relaying the message.

Given the two roles for an ADMD, this subsection describes two distinct sets of functional requirements for an ADMD. The first is the relaying requirement that is needed to provide PRMD and other ADMD interworking. The second is analogous to the PRMD's support to its customers through the integrated UAs. These are distinct functional differences. The services provided to the UAs of an ADMD are independent of the requirement that an ADMD provide a function for interworking with any type of Management Domain (MD). Figure 13.4.1 illustrates the two roles played by an ADMD.

This section is presented in the form of deviations from the agreements applicable to PRMD-to-PRMD (section 13.3.0). Unless explicitly noted in the remainder of this section, all of the specifications for PRMD to PRMD apply to PRMD to ADMD and ADMD to ADMD.



(a)

(b)

Figure 13.4.1 An ADMD may (b) or may not (a) serve as a relay.

13.4.2 ADMDs as Relays

The following apply to an ADMD when it serves as a relaying domain.

1) ADMDs will relay all content types (not just P2) unchanged in the absence of a request for conversion.

2) P1 Protocol Change

Protocol Elements	Class	Comments
DeliveryReportContent		
intermediate Trace Information	G	If requested by other than the originating domain, return of the desired information cannot be assured, as the return path may be different and exclude that domain.
DeliveredInfo		
typeOfUA	H	This element must be generated by PRMDs with value of "PRIVATE".
ReportedRecipientInfo		
SupplementaryInformation	H	Domains providing access to TELEX/TELETEX recipients, whether directly or indirectly as a result of bilateral agreements between domains, must ensure that this information, when present, is accessible by the recipient of the delivery report.
GlobalDomainIdentifier		
PrivateDomainIdentifier	R	For PRMDs. ADMDs shall take appropriate actions upon receipt of this element.

3) O/R Names

O/R Names shall consist of:

CountryName
AdministrationDomainName

as well as one of the following:

PrivateDomainName
PersonalName
OrganizationName
OrganizationalUnit
UniqueUAIdentifier
X121Address

and permits the optional inclusion of a

DomainDefinedAttributeList

Note that the destination PrivateDomainName or OrganizationName must be present if destined for a PRMD. The ADMD relaying the message to that destination PRMD requires this element.

P1 Originator Name

Management Domains (MDs) must specify in the ADMD name field of the O/R Name StandardAttributeList in P1, the name of the administration domain:

(a) to which the message is being sent (in recipient names)
(b) from which the message originated (in the originator name).

13.4.3 Interworking with Integrated UAs

If the message originates at a UA owned by an ADMD, or is delivered to such a UA, the O/R Name follows the same Form 1 Variant 1 constraints as the base specifications; except that the ADMD name is the name of the owning ADMD and instead of supplying a PRMD Name, one (or more) of the following must be provided:

OrganizationName
OrganizationalUnit
PersonalName

and may optionally include a

DomainDefinedAttributeList

13.4.4 Differences with other Profiles

13.4.4.1 NTT Profile

There are no outstanding issues regarding interworking between NTT-conformant systems and NBS-conformant systems with the exception of the number of recipients. The Extension Identifier field may contain a maximum value of 32K-1; however, according to the current NTT profile, if a message with more than 256 recipients is received, the NTT-conformant domain will generate a nondelivery notification. This also applies to the ReportedRecipientInfo in a delivery report.

13.4.4.2 CEPT Profile

For further study.

13.4.5 Connection of PRMDs to Multiple ADMDs.

Given that Management Domain names (both PRMD and ADMD) shall be unique within the U.S., then when an ADMD is presented a message for transfer from a PRMD, it will accept O/R Names (both originator and recipient) which have an AdministrationDomainName field value different than the administration's name. "Accept" implies the attempt to route/deliver the message shall be made, as appropriate, based upon the knowledge that MD names are unique.

Whether this functionality is required by an administration for conformance to this agreement is for further study.

If a PRMD is connected to two or more ADMDs which are not effectively connected (either directly or via a third ADMD), full X.400 functionality shall not be available. Problems occur especially in the areas of:

-Naming
-Routing
-Replying.

13.4.6 Connection of an ADMD to a Routing PRMD

It is possible for a collection of interconnected private domains to establish one domain as the "gateway" to an ADMD, and hence to the world.

If an ADMD is connected to such a gateway PRMD, the individual private domains shall be registered with the administration. Administrations need not support such connections.

Note also that upon receipt by the ADMD of a message originating somewhere within the PRMD collection, that the TraceInformation may contain more than one element.

13.4.7 Management Domain Names

All Management Domain Names (both Private and Administration) shall be unique within the U.S.

A central naming authority shall be established to register domain names.

13.4.8 Envelope Validation Errors

ADMDs will validate P1 Envelopes in the following areas:

a. The X.409 syntax of all elements should be checked.

b. The pragmatic constraint limits (lengths of fields and number of occurrances of fields) should be checked.

c. Semantic validation of the following selected set of elements should be done: originator O/R Name, original EncodedInformationTypes (but not against the actual contents), Priority, PerMessageFlag, and RecipientInfo.

   Validation of the MPDUIdentifier is for further study.

   For relaying messages, TraceInformation will be examined to detect looping route problems. Additional validation of the TraceInformation is for further study.

The actual ReasonCodes and DiagnosticCodes to be returned in the case of validation failure are for further study.

13.4.9 For further study

Among issues reserved for further study are:

1) RTS password management
2) Billing
3) Quality of service
4) Diagnostic information in support of operations
5) Intra-Domain Routing
6) Multi-Vendor Domains.

RTS password management is currently believed to be a local matter.

13.5 ERROR REPORTING

This section describes appropriate actions to be taken by non-relaying domains upon receipt of protocol elements which are not supported in this profile, malformed MPDUs, unrecognized O/R Name forms, content errors, errors in reports, and unexpected values for protocol elements.

13.5.1 MPDU Encoding

The MPDU should have a context-specific tag of 0, 1, or 2. If it does not have one of these tags, it is not possible to figure out who originated the message. Therefore, the way this error is reported is a local matter.

13.5.2 Contents

Once delivery to the UA has occurred, it is not possible to report errors in P2 information to the originator. In addition, it seems unreasonable to insist that the MTA that delivers a message ensures that the P2 content of the message is okay. As a result, the handling of content errors is a local matter.

13.5.3 Envelope

13.5.3.1 Pragmatic Constraint Violations

In all cases of pragmatic constraint violation, a nondelivery report should be generated with a ReasonCode of unableToTransfer, and a DiagnosticCode of invalidParameters. Note: it would be desirable for the CCITT to add a DiagnosticCode of pragmaticConstraintViolation to allow a more meaningful description of this problem. A request for this new DiagnosticCode has been submitted.

13.5.3.2 Protocol Violations

If not all required protocol elements are present, a nondelivery report with a ReasonCode of unableToTransfer and a DiagnosticCode of invalidParameters should be generated. Note: it would be desirable for the CCITT to add a DiagnosticCode of protocolViolation to allow a more meaningful description of this problem. A request for this new DiagonsticCode has been submitted.

If a protocol element is expected to be of one type, but is encoded as another, then a nondelivery report with a ReasonCode of unableToTransfer and a DiagnosticCode of invalidParameters should be generated. Note: it would be desirable for the CCITT to add a DiagnosticCode of protocolViolation to allow a more meaningful description of this problem. A request for this new DiagnosticCode has been submitted.

13.5.3.3 O/R Names

The domain that has responsibility for delivering a message should also have the responsibility to send the nondelivery notification if the message cannot be delivered. Therefore, each domain should only validate the O/R Names of recipients with responsibility flags set to TRUE. In addition, a nondelivery notification can only be sent if the originator's O/R Name is valid.

If any element in the O/R Name is unrecognized or if the CountryName, AdministrationDomainName, and one of PrivateDomainName and OrganizationName are not all present, then a nondelivery report should be generated with a ReasonCode of unableToTransfer, and a DiagnosticCode of unrecognizedORName. If the message can be delivered even though the ORName is invalid, delivery is a local matter. Note, however, that if the message is delivered, the invalid ORName might be propagated through the X.400 system (e.g., by forwarding).

If the O/R Name has all of the appropriate protocol elements and the message still cannot be delivered to the recipient, the following DiagnosticCodes may appear in the nondelivery report: unrecognizedORName, ambiguousORName, and uaUnavailable.

13.5.3.4 TraceInformation

Since non-relaying domains need not do loop suppression, domains with responsibility for delivering the message need not be concerned about the semantics of the TraceInformation, that is, arrival time and converted EncodedInformationTypes can be provided to the UA without inspection by the MTAs of the domain as long as the TraceInformation is properly encoded according to X.409.

13.5.3.5 Unsupported X.400 Protocol Elements

The protocol elements defined in X.400 but unsupported by this profile are: the deferredDelivery and PerDomainBilateralInfo parameters of the UMPDUEnvelope, the ExplicitConversion parameter of RecipientInfo, and the alternateRecipientAllowed and contentReturnRequest bits of the PerMessageFlag. Appropriate actions are described below for domains that do not support the protocol elements.

13.5.3.5.1 deferredDelivery

The domain shall do one of the following:

- deliver at once,
- hold for deferred delivery,
- return a nondelivery notification with a ReasonCode of unableToTransfer.

Note: it would be desirable for the CCITT to add a diagnostic code of noBilateralAgreement to allow a more meaningful description of this problem. A request for this new diagnostic code has been submitted.

13.5.3.5.2 PerDomainBilateralInfo

If a domain receives this service element, the service element can be ignored, and the message should be delivered if possible.

13.5.3.5.3 ExplicitConversion

If ExplicitConversion is requested the message should be delivered if possible. That is, if the UA is registered to accept the EncodedInformationTypes of the message, then the message should be delivered even though the domain could not perform the requested conversion. If delivery is not possible, then a nondelivery report should be generated with a ReasonCode of conversionNotPerformed with no DiagnosticCode.

13.5.3.5.4 alternateRecipientAllowed

If a domain receives this service element the service element can be ignored, and this message should be delivered if possible.

13.5.3.5.5 contentReturnRequest

If a domain receives this service element, the service element can be ignored, and the message should be delivered if possible.

13.5.3.6 Unexpected Values for INTEGER Protocol Elements

There are three INTEGERs in the P1 Envelope. Appropriate actions are described below for domains receiving unexpected values for Priority, ExplicitConversion, and ContentType.

13.5.3.6.1 Priority

Additional values for Priority have been suggested by at least one group of implementors as upward compatible changes to the X.400 Recommendations. Therefore, if a domain receives an unexpected value for Priority, and this value is greater than one byte in length, a nondelivery report should be generated with a ReasonCode of unableToTransfer and DiagnosticCode of invalidParameters. If the value is less than or equal to one byte, the domain can either generate a nondelivery report as previously specified or default the Priority to normal and deliver the message.

13.5.3.6.2 ExplicitConversion

The message should be delivered if possible. That is, if the UA is registered to accept the EncodedInformationTypes of the message, then the message should be delivered even though the domain could not perform the requested conversion. If delivery is not possible, then a nondelivery report should be generated with a ReasonCode of conversionNotPerformed with no DiagnosticCode.

13.5.3.6.3 ContentType

If the ContentType is not supported, then a nondelivery report should be generated with a ReasonCode of unableToTransfer, and a DiagnosticCode of invalidParameters. Note: it would be desirable for the CCITT to add a DiagnosticCode of contentNotSupported to allow a more meaningful description of this problem. A request for this new DiagnosticCode has been submitted.

13.5.3.7 Additional Service Elements

In the absence of bilateral agreements to the contrary, receipt of privately tagged elements and protocol elements in addition to those defined in X.400 will result in a nondelivery report with a ReasonCode of unableToTransfer and a DiagnosticCode of invalidParameters.

13.5.4 Reports

There is no mechanism for returning a delivery or status report due to errors in the report itself. Therefore the handling of errors in reports is a local matter.

13.6 MHS USE OF DIRECTORY SERVICES

Recommendation X.400 recognizes the need of MHS users for a number of directory service elements. Directory service elements are intended to assist users and their UAs in obtaining information to be used in submitting messages for delivery by the MTS. The MTS may also use directory service elements to obtain information to be used in routing messages. Some functional requirements of directories have been identified and are listed below.

1. Verify the existence of an O/R name.
2. Return the O/R address that corresponds to the O/R name presented.
3. Determine whether the O/R name presented denotes a user or a distribution list.
4. Return a list of the members of a distribution list.
5. When given a partial name return a list of O/R name possibilities.
6. Allow users to scan directory entries.
7. Allow users to scan directory entries selectively.
8. Return the capabilities of the entity referred to by the O/R name.
9. Provide maintenance functions to keep the directory up-to-date.

In addition to functionality, a number of operational aspects must be considered. These include user-friendliness, flexibility, availability, expandability, and reliability.

Currently, these aspects of directory service elements and procedures are under study by both the CCITT and the ISO. Both organizations are committed to the development of a single Directory Service specification for use by MHS and all other OSI based applications.

Given the incomplete nature of the ongoing activities within the CCITT and the ISO, no implementation details will now be provided for MHS use of Directory Services. Implementation agreements for MHS Use of Directory Services will be issued when current activities within the CCITT and the ISO are stable.

It is recognized that these agreements enable a wide variety of naming and addressing attributes (see section 13.3.5.3.3 ORName Protocol Elements) wherein each PRMD may adopt particular routing schemes within its domain. These agreements make no attempt to recommend a standard practice for electronic mail addressing.

Inter-PRMD addressing may be secured according to practices outside the scope of these agreements, such as:

- manual directories
- on-line directories
- ORName address specifications
- ORName address translation

Further, each PRMD may adopt naming and addressing schemes wherein the user view may take a form entirely different from the attributes reflected in section 13.3.4.2.2 herein. And, each PRMD may have one user view for the originator form and another for the recipient form, and perhaps other forms of user addressing. In some cases (e.g., receipt notifocation) these user forms must be preserved within the constraints of these implementation agreements. However, mapping between one PRMD user form to another PRMD user form, via the X.400 ORName attributes of these agreements, is outside the scope of these agreements.

13.7 CONFORMANCE

In order to ensure that products conform to these implementation agreements, it is necessary to define the types and degrees of conformance testing products that must pass before they may be classified as conformant. This section defines the conformance requirements and provides guidelines for the interpretation of the results from this type of testing.

In order to achieve a minimum level of confidence in the conformance of a product, the most basic requirements a product must meet to be classified as conformant to these agreements are provided. This minimal set was defined to ensure that the resulting MHS network will provide a reasonable interpersonal messaging facility to its users while still giving a reasonable assurance that conforming products can soon be made available. In addition, the full conformance requirements for products implementing all aspects of X.400 Messaging governed by these agreements are provided.

This section is incomplete and will be enhanced in future versions of this agreement. Later versions will reflect the problems of conformance testing and will outline specific practices and recommendations to aid the development of conformance tests and procedures.

13.7.1 Definition of Conformance

For this section, the term conformance is defined by the following:

1. The tests indicated for this section are intended to establish a high degree of confidence in a statement that the implementation under test (IUT) conforms (or does not conform) to the agreements of this section.

2. Conformance to a service element means that the information associated with the service element is made accessible to the user (person or process) whenever this agreement says that this information should be available.

   Accessible means that information must be provided describing how a user (person or process):

   a) causes appropriate information to be displayed, or

   b) causes appropriate information to be obtained.

3. Conformance to P1, P2, and RTS as part of an X.400 OSI application requires that only the external behavior of that OSI system adheres to the relevant protocol standards.

   In order to achieve conformance to this section, it is not required that the inter-layer interfaces be available for testing purposes.

4. Conformance to the protocols requires:

   a) that MPDUs correspond to instances of syntactically correct data units,

   b) MPDUs in which the data present in the fields and the presence (or absence) of those fields is valid in type and semantics as defined in X.400, as qualified by this profile,

   c) correct sequences of protocol data units in responses (resulting from protocol procedures).

5. Statements regarding the conformance of any one implementation to this profile are not complete unless a Protocol Implementation Conformance Statement (PICS) is supplied.

6. The term "Implementation Under Test" (IUT) is interchangeable with the term "system" in the definition of conformance, and may refer to:

   a) a domain, which may be one or more MTA's with co-located or remote UA's,

   b) a single instance of an MTA and co-located UA with X.400 (P1, P2, RTS and session) software,

   c) a relaying product with P1, RTS and session software,

   d) a gateway product.

7. Tests for conformance apply independently to:

   a) origination,

   b) reception,

   c) relaying.

13.7.2 Conformance Requirements

Conformance to this specification requires that all the services listed as supported in sections 13.3 and 13.4 of these agreements are supported in the manner defined, in either the CCITT X.400 Recommendations or these agreements.

It is the intention to adopt, where and when appropriate the testing methodology and/or the abstract test scenarios currently being defined by the CCITT X.400 Conformance Group. However it is recognized that formal CCITT Recommendations relating to X.400 Conformance Testing will not be available until 1988.

13.7.2.1 Initial Conformance

This section is intended to provide guidelines to vendors who envisage having X.400 products available prior to any formal mechanism, or 'Conformance Test Center' being made accessible that would allow for conformance to this product specification to be tested.

It is feasible that vendors and carriers will want to enter bilateral test agreements that will allow for initial trials to be carried out for the purposes of testing initial interworking capabilities. It is equally feasible that for the purposes of testing interoperability, only a subset of this specification will initially be tested. Therefore it is recommended that the following subset of total information be made accessible to allow for meaningful testing.

Note: By claiming conformance to this subset of information the vendor or carrier cannot claim conformance to this entire specification.

There are two aspects to the requirements, interworking and service, as described in the following sections.

13.7.2.1.1 Interworking

The interworking requirements for conformance implies that tests be done to check for the syntax and semantics of protocol data elements for a system as defined by their classifications (i.e., X, H, G, R, and M). For an origination system, this implies that the protocol elements generated must be correct. For a relay system, the correct protocol elements should be relayed as appropriate. And for a recipient system, a message with correct protocol elements must not be rejected where appropriate.

13.7.2.1.2 Service

For information available to the recipients via the IPMessage Heading and Body, the following should be made accessible:

- IPMessage ID - only the PrintableString portion of the IPMessageId needs to be accessible.
- subject
- primaryRecipients
- copyRecipients
- blindCopyRecipients
- authorizingUsers
- originator
- inReplyTo
- replyToUsers
- importance
- sensitivity
- IA5Text BodyPart

14. DIRECTORY SERVICES PROTOCOLS

The directory services protocols' implementation specifications are being prepared by the DS SIG.

15. PERFORMANCE

To be completed.

16. SECURITY

To be completed.

REFERENCES

NBS

1. FIPS 107, Local Area Networks: Baseband Carrier Sense Multiple Access with Collision Detection Access Method and Physical Layer Specifications and Link Layer Protocol, NTIS, U.S. Department of Commerce, 5285 Port Royal Road, Springfield, VA 22161.

2. FIPS 100, Interface Between Data Terminal Equipment (DTE) and Data Circuit-Terminating Equipment (DCE) For Operation With Packet-Switched Data Communications Networks, NTIS, U.S. Department of Commerce, 5285 Port Royal Road, Springfield, VA 22161.

3. ICST/SNA-85-10, Implementation Agreements Among Participants of OSINET, edited by Jerry Mulvenna, National Bureau of Standards.

IEEE

1. IEEE Project 802
   Local Area Network Standards
   P802.2 Logical Link Control
   November, 1982.

2. IEEE Project 802
   Local Area Network Standards
   IEEE Standard 802.4
   Token - Passing Bus Access Method
   and
   Physical Layer Specification.

3. IEEE Project 802
   Local Area Network Standards
   IEEE Standard 802.3
   CSMA/CD Access Method
   and
   Physical Layer Specification.

The above documents may be obtained from: IEEE Standards Office, 345 East 47th Street, New York, N.Y. 10017.

ISO

1. Addendum to DIS 8473 Covering Provision of the Connectionless-Mode Subnetwork Service, ISO/TC97/SC 6/N3453.

2. Network Service Definition, DIS 8348, ISO/TC97/SC6 N2990.

3. Addendum to the Network Service Definition Covering Connectionless Data Transmission, DIS 8348 DAD1, N3152.

4. Addendum to the Network Service Definition Covering Network Layer Addressing, DP 8348 DAD2, N3134.

5. Internal Organization of the NetworkLayer, WD, N3141.

6. Protocol for Providing the Connectionless Network Service, DIS 8473, N3154.

7. Information Processing Systems - Open Systems Interconnection - Transport Service Definition, ISO IS8072, 1984.

8. Information Processing Systems - Open Systems Interconnection - Transport Protocol Specification, ISO IS8073, 1984.

9. Information Processing Systems - Open Systems Interconnection - Session Service Definition, ISO DIS8326, 1984.

10. Information Processing Systems - Open Systems Interconnection - Session Protocol Specification, ISO DIS8327, 1984.

11. Information Processing Systems - Open Systems Interconnection - File Transfer Access and Management Part 1: General Description, ISO DP8571/1, TC97/SC16 N 1669, February 1984.

12. Information Processing Systems - Open Systems Interconnection - File Transfer Access and Management Part II: The Virtual Filestore, ISO DP 8571/2, TC97/SC16 N1670, February 1984.

13. Information Processing Systems - Open Systems Interconnection - File Transfer Access and Management Part III: Service Definition, ISO DP8571/3, TC97/SC16 N1671, February 1984.

14. Information Processing Systems - Open Systems Interconnection - File Transfer Access and Management Part IV: Protocol Specification, ISO DP8571/4, TC97/SC16 N1672, February 1984.

15. Data Communication - X.25 Packet Layer Specification for Data Terminal Equipment, ISO/TC 97/SC 6 N 2641, ISO/DP 8208, 1983.

16. 7-bit Coded Character Set for Information Processing Interchange, ISO-646, 1973.

17. Information Interchange--Representation of Local Time Differentials, ISO-3307, 1975.

18. Draft Network Layer Management Protocol for the exchange of routing information between end systems and intermediate systems ISO/TC97/SC6/3862 January 1986.

19. Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management Part I: General Description, ISO DIS8571/1, TC97/SC21 N2371, August 1986.

20. Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management Part II: The Virtual Filestore, ISO DIS8571/2, TC97/SC21 N2372, August 1986.

21. Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management Part III: File Service Definition, ISO DIS8571/3, TC97/SC21 N2373, August 1986.

22. Information Processing Systems - Open Systems Interconnection - File Transfer, Access and management Part IV: File Protocol Specification, ISO DIS8571/4, TC97/SC21 N2374, August 1986.

23. Information Processing Systems - Open Systems Interconnection - Connection-Oriented Presentation Service Definition, ISO DIS8822, TC97/SC21 N1594, May 1986.

24. Information Processing Systems - Open Systems Interconnection - Connection-Oriented Presention Protocol Specification, ISO DIS8823, TC97/SC21 N1594, May 1986.

25. Information Processing Systems - Open Systems Interconnection - Service Definition for Common Application Service Elements - Part 2: Association Control, ISO DIS8649/2, TC97/SC21 N1493, May 1986.

26. Information Processing Systems - Open Systems Interconnection - Protocol Specification for Common Application Service Elements Part 2: Association Control, ISO DIS8650/2, TC97/SC21 N1494, May 1986.

The above documents may be obtained from:

Frances E. Schrotter
ANSI
ISO TC97/SC6 Secretariat
1430 Broadway
New York, N.Y. 10018

CCITT

1. X.400 (Red Book, 1984), Message Handling Systems: System Model-Service Elements.

2. X.401 (Red Book, 1984), Message Handling Systems: Basic Service Elements and Optional User Facilities.

3. X.408 (Red Book, 1984), Message Handling Systems: Encoded Information Type Conversion Rules.

4. X.409 (Red Book, 1984), Message Handling System: Presentation Transfer Syntax and Notation.

5. X.410 (Red Book, 1984), Message Handling System: Remote Operations and Reliable Transfer Server.

6. X.411 (Red Book, 1984), Message Handling Systems: Message Transfer Layer.

7. X.420 (Red Book, 1984), Message Handling Systems: Interpersonal Messaging User Agent Layer.

8. X.430 (Red Book, 1984), Message Handling Systems: Access Protocol for Teletex Terminals.

9. X.215 (Red Book, 1984), Session Service Definition for Open Systems Interconnection for CCITT Applications.

10. X.225 (Red Book, 1984), Session Protocol Specification for Open Systems Interconnection for CCITT Applications.

11. X.400 - Series Implementor's Guide (Version 3, 1986).

The above documents may be obtained from: International Telecommunications Union, Place des Nations, CH 1211, Geneve 20 SWITZERLAND.

Miscellaneous

1. [Edge 84] S. W. Edge, An Adaptive Timeout Algorithum for Retransmission Across a Packet Switching Network, ACM Computer Communications Review, Vol. 14, No. 2, June 1984.

2. [Jain 85] R. Jain, Divergence of Timeout Algorithms for Packet Retransmission, Proceedings IEEE Computer Communications Conference, Phoenix March 28-29, 1986.

3. [Mill 83] D. L. Mills, Internet Delay Experiments, DARPA Network Working Group RFC #889, December 1983.

APPENDIX A: INTERPRETATION OF SERVICE ELEMENTS

The work on service element definitions is limited to those that are defined as 'supported' in section 13.3 of this specification. Furthermore it is not the intent of this section to define how information should be made available or presented to a MHS user, nor is it intended to define how individual vendors should design their products. In addition, statements on conformance to a specific service element and the allocation of error codes that are generated as a result of violations of the service should be defined in the sections on conformance and errors as part of the main product specification. The main objective is to provide clarification, where required, on the functions of a service element, and in particular what the original intent of the Recommendations were.

SERVICE ELEMENTS

The following Service Elements defined in X.400 have been examined and require further text to be added to their definitions to represent the proposed implementation of these service elements by the X.400 SIG.

The service element clarifications are to be taken in the context of this profile.

Service elements not referenced in this section are as defined in X.400.

PROBE

A PRMD need <u>not</u> generate probes.

If a probe is addressed to and received by a PRMD, the PRMD <u>must</u> respond with a <u>Delivery Report</u> as appropriate at the time the probe was processed.

DEFERRED DELIVERY

In the absence of bilateral agreements to the contrary, Deferred Delivery and Deferred Delivery Cancellation are local matters (i.e., confined to the originating domain) and need <u>not</u> be provided.

The extension of Deferred Delivery beyond the boundaries of the initiating domain is via bilateral agreement as specified in Section 3.4.2.1 of X.411.

Content Type Indication

It is required that both an originating and recipient domain be able to support P2 content type. The ability for domains to be able to exchange content types other than P2 will depend on the existence of bilateral or multi-lateral agreements.

Original Encoded Information Types Indication

It is required that both an originating and recipient domain be able to support IA5 text. Support for other encoded information types, for the purposes of message transfer between domains, will depend on the existence of bilateral or multi-lateral agreements.

The use of the 'unspecified' form of encoded information type should only be used when the UMPDU content represents an SR-UAPDU or contains an auto-forwarded IM-UAPDU.

The original encoded information type of a message is not meaningful unless a message is converted en route to the recipient. These agreements support only IA5 text, which should not undergo conversion. The original encoded information types should be made accessible to the recipient for upward compatibility with the use of non-IA5 text message body parts.

Registered Encoded Information Types

A UMPDU with an 'unspecified' value for Original Encoded Information Type shall be delivered to the UA.

Delivery Notification

The UAContentID may be used by the recipient of the delivery notification for correlation purposes.

Disclosure of Other Recipients

This service is not made available by originating MTAE's to UAE's, but must be supported by relaying and recipient MTAE's.

By supporting the disclosure of other recipients the message recipient can be informed of the O/R names of the other recipient(s) of the message, as defined in the P1 envelope in addition the O/R Descriptors within the P2 header.

These agreements do not support initiation of disclosure of other recipients, but the information associated with it should be made accessible to the recipient for upward compatibility with support for the initiation of this service element.

Typed Body

As defined in X.400 with the addition of the Private Body Types that are to be supported. At present there is no mechanism provided within X.420 that would allow you to respond to reception of an unsupported body type.

Action taken in this situation is a local matter.

Blind Copy Recipient Indication

It should be considered that the recipient's UA acts on behalf of the recipient, and therefore may choose to disclose all BCC recipients to each other. Therefore it is the responsibility of the originating domain to submit two or more messages, depending on whether or not each BCC should be disclosed to each other BCC.

Auto-Forwarded Indication

A UA may choose not to forward a message that was previously auto-forwarded. In addition there is no requirement for an IPM UA that does not support non-receipt or receipt notification to respond with a non-receipt notification when a message is auto-forwarded.

Primary and Copy Recipients Indication

It is required that at least one primary recipient be specified; however, for a forwarded message this need not be present. The recipient UA should be prepared to accept no primary and copy recipients to enable future interworking with Teletex, Fax, etc.

Sensitivity Indication

A message originator should make no assumptions as to the semantic interpretation by the recipients UA regarding classifications of sensitivity. For example, a personal message may be printed on a shared printer.

Reply Request Indication

In requesting this service an originator may additionally supply a date by which the reply should be sent and a list of the intended recipients of the reply. If no such list is provided than the initiator of the reply sends the reply to the originator of the message and any recipients the reply initiator wishes to include. The replytoUsers and the replyBy date may be specified without any explicit reply being requested. This may be interpreted by the recipient as an implicit reply request. Note that for an auto-forwarded message an explicit or implicit reply request may not be meaningful.

Body Part Encryption

The original encoded information type indication includes the encoded information type(s) of message body parts prior to encryption by the originating domain. The ability for the recipient domain to decode an encrypted body part is a local matter. Successful use of this facility can only be guaranteed if there exists bilateral agreements to support the exchange of encrypted body parts.

Forwarded IP-message Indication

The following use of the original encoded information type in the context of forwarded messages is clarified:

- If forwarding a private message body part the originator of the forwarded message shall set the original encoded information types in the P1 envelope to undefined for that body part.
- The encoded information types of the message being forwarded should be reflected in the new original encoded information types being generated.
- See Appendix B on recommended practices for the use of the delivery information as part of Forwarded IP-message.

Multipart Body

It is the intent of multipart bodies to allow for the useful and meaningful structuring of a message that is constructed using differing body part types. For example, it is not recommended that a message made up of only IA5 text should be represented as a number of IA5 body parts, each one representing a paragraph of text.

APPENDIX B: RECOMMENDED PRACTICES

B.1 RECOMMENDED PRACTICES IN P2

a) ORDescriptor

Vendors following the NBS/OSI Workshop guidelines shall, whenever possible, generate the ORName portion of an ORDescriptor in ALL IPM Heading fields.

b) ForwardedIPMessage BodyParts

ForwardedIPMessage BodyParts should be nested no deeper than eight. There is no restriction on the number of ForwardedIPMessage BodyParts at any given depth.

c) DeliveryInformation

It is strongly recommended that DeliveryInformation be supplied in both forwarded and autoforwarded message body parts. DeliveryInformation is useful when a message has multiple forwarded message body parts because without it, the EncodedInformationType(s) of the component forwarded messages cannot be deduced easily. DeliveryInformation is useful for autoforwarded messages because the EncodedInformationType of an autoforwarded message is "unspecified" and the EncodedInformationType(s) of the message cannot be determined easily without it. Absence of the EncodedInformationType(s) makes it difficult for a UA to easily determine whether the message can be rendered.

B.2 RECOMMENDED PRACTICES IN RTS

a) The calling party Network address should be used for MTA validation rather than the mTAName and password. The calling party address validation method is preferable since the recover function (in S-CONNECT) does not allow the specification of mTAName and password.

b) In the case where S-U-ABORT indicates a temporaryProblem, re-establishment of the session should not be attempted for a "sensible" time period (typically not less than five minutes).

In instances where this delay is not required or necessary, report a localSystemProblem.

c) S-U-EXCEPTION-REPORT reason codes can be interpreted as follows:

- receiving ability jeopardized (value 1)
  Possible meaning: The receiving RTS knows of an impending system shutdown.

- local ss-User error (value 5)
  Possible meaning: <for further study>

- unrecoverable procedure error (value 6)
  Possible meaning: the current activity is NOT recoverable.

- non specific error (value 0)
  Possible meaning: <for further study>

- sequence error (value 3): The S-ACTIVITY-RESUME request specified a minor synchronization point serial number which does not match the checkpoint data.

B.3 RECOMMENDED PRACTICES WITH X.409

The following practices are recommended for use with X.409.

a. The maximum length of a primitive data element is 256.

b. Bit Strings should be built using primitive form. The constructor form should not be used except in the case of very long Bit Strings (e.g., 63Fax or Voice).

c. All defined bits of a Bit String should be present.

   - Note that, in accordance with X.409, defined bits need not be present; missing bits are assumed to be zero.

   - To ensure upward compatibility, Bit Strings of excess length must also be allowed; the excess bits are ignored.

d. The maximum definite length should be (2**32)-1. <For further study>

e. It is intended that implementations support upwardly compatible changes to X.409, as defined in Version 3 of the X.400-Series Implementor's Guide, but no guarantees will be made about initial implementations.

f. The concrete encoding of ANY must be a valid X.409 type, and can only be omitted if it is an OPTIONAL element in a SET or SEQUENCE.

B.4 RECOMMENDED PRACTICES FOR ORName

Table 13.3.8 stipulates that the StandardAttributeList must contain either PrivateDomainName or Organization Name. It is recommended that, for both originator and recipients in a private domain, the PrivateDomainName field be used.

It is recommended that there should be a DDA to be used in addresing UAs in existing mail systems, in order to curtail the proliferation of different types of DDAs used for the same purpose. The syntax of this DDA conforms to the CCITT Pragmatic Constraints, and thus has a maximum value length of 128 octets and a type length of 8 octets, each of type Printable String. One occurrance!

This DDA has the type name "ID" (in uppercase). It contains the unique identifier of the UA used in addressing within the domain. This DDA is to be exclusively used for routing within the destination domain (i.e. once routed to that domain via the mandatory components of the Standard Attribute list); any other components of the Standard Attribute list may be provided. If they conflict delivery is not made.

The contents of the valve parameter need not be validated in the originating domain or any relaying domain, but simply transferred intact to the next MTA/domain.

APPENDIX C: RENDITION OF IA5Text AND T61String CHARACTERS

C.1 GENERATING AND IMAGING IA5Text

The characters that may be used in an IA5String are the graphic characters (including Space), control characters and Delete of the IA5 character repertoire ISO 646.

The graphic characters that may be used with a guaranteed rendition are those related with positions 2/0 to 2/2, 2/5 to 3/15, 4/1 to 5/10, 5/15 and 6/1 to 7/10 in the basic 7-bit code table.

The other graphic characters may be used but have no guaranteed rendition.

The control characters that may be used but have no guaranteed effect are a subset consisting of the format effectors 0/10 (LF), 0/12 (FF) and 0/13 (CR) provided they are used in one of the following combinations:

CR LF to start a new line
CR FF to start a new page (and line)
LF .. LF to show empty lines (always after one of the preceding combinations).

The other control characters or the above control characters in different combinations may be used but have no guaranteed effect.

The character Delete may occur but has no guaranteed effect.
The IA5String in a P2 IA5Text BodyPart represents a series of lines which may be divided into pages. Each line should contain from 0 to 80 graphic characters for guaranteed rendition. Longer lines may be arbitrarily broken for rendition. Note that X.408 states that for conversion from IA5Text to Teletex, the maximum line length is 77 characters.

C.2 GENERATING AND IMAGING T61String

For further study.

APPENDIX D: FTAM DOCUMENT TYPES

Part 1: Document Types
Part 2: Constraint Sets
Part 3: Abstract Syntaxes
Part 4: Transfer Syntaxes

Part 1: Document Types

Entry Number: NBS-1

Document_Type_Name:

{ISO registration-authority NBS FTAM( ) document(6) UNDEF(0)}

Document_Descriptor Value: unstructured binary file

Document_Semantics:

The document consists of a single data unit. The data unit consists of an unbounded sequence of data elements. Each data element is an octet string.
Note: The boundaries between transferred data elements are not maintained
by the filestore.

Scope and Field of Application:

This document type defines the contents of a file for storage and for transfer using FTAM.

Constraint_Set_Name:

{ISO standard 8571 constraint set name(5) unstructured(1)}

Abstract_Syntax:

The abstract syntax of each Data Element is an instance of the ASN.1 data type OctetString.

Abstract_Syntax Name:

{ISO registration-authority NBS FTAM( ) abstract syntax(2) NBS-AS1(0)}

Transfer_Syntax:

An implementation supporting this abstract syntax shall support a transfer syntax for each data element obtained by applying the ASN.1 Basic Encoding Rules (ISO 8825) to the data element and concatenating the resulting octets.

Note: This transfer syntax is not self delimiting.

Implementations may optionally support other named transfer syntaxes for this abstract syntax.

Transfer_Syntax Name:

{ISO registration-authority NBS FTAM( ) transfer syntax(3) NBS-TS1(0)}

Concatenation:

Concatenation of this type with itself is possible, and produces a document of the same type consisting of one data unit which is the concatenation of the octet string(s) from one file with the octet string(s) of the other file.

Note: The boundary of the original octet string(s) is no longer visible.

Simplifications:

A document of this type cannot be accessed as any other document type.

Entry Number: NBS-2

Document Type Name:

{ISO registration-authority NBS FTAM( ) document(6) VARCRLF(1)}

Document Descriptor Value: unstructured text file

Document Semantics:

The document consists of a single data unit. The data unit consists of an unbounded sequence of data elements. Each data element is an IA5String. The last two characters of each data element are carriage return followed by line feed. Neither the character carriage return nor the character line feed may appear elsewhere in the data element.

Note: The boundaries between transferred data elements are not maintained by the filestore.

Scope and Field of Application:

The document type defines the contents of a file for transfer using FTAM.

Note that this document type should only be used for transferring entire text files in the case where NBS-4 is not supported. It has an implicit structure which allows, for example, text files stored in UNIX format (lines terminated by LF) to be converted to a format in which lines are terminated by CR followed by LF and vice versa.

Constraint Set Name:

{ISO standard 8571 constraint set name(5) unstructured(1)}

Abstract Syntax:

The abstract syntax of each Data Element is an instance of the ASN.1 data type IA5String.

Abstract Syntax Name:

{ISO registration-authority NBS FTAM( ) abstract syntax(2) NBS-AS1(0)}

Transfer Syntax:

An implementation supporting this abstract syntax shall support a transfer syntax (for each data element obtained by applying the ASN.1 Basic Encoding Rules (ISO 8825) to the data element and concatenating the resulting octets.

Note: This transfer syntax is not self-delimiting.

Implementations may optionally support other named transfer syntaxes for this abstract syntax.

Transfer Syntax Name:

{ISO registration-authority NBS FTAM( ) transfer syntax(3) NBS-TS1(0)}

Concatenation:

Concatenation of this type with itself is possible, and produces a document of the same type consisting of one data unit which is the concatenation of the octet string(s) from one file with the octet string(s) of the other file.

Note: The boundary of the original octet string(s) is no longer visible.

Simplification:

A document of this type can be accessed as a document of type NBS-1 by specifying a document type of NBS-1 in the Contents Type parameter of the F-OPEN request, and limiting access context to US on F-READ.

Entry Number: NBS-3

Document Type Name:

{ISO registration-authority NBS FTAM( ) document(6) 8859VARCRLF(2)}

Document Descriptor Value: unstructured text file

Document Semantics:

The document consists of a single data unit. The data unit consists of an unbounded sequence of data elements. Each data element is an 8859String. The last two characters of each data element are carriage return followed by line feed. Neither the character carriage return nor the character line feed may appear elsewhere in the data element.

Note: The boundaries between transferred data elements are not maintained by the filestore.

Scope and Field of Application:

The document type defines the contents of a file for transfer using FTAM.

Note that this document type should only be used for transferring entire text files in the case where NBS-5 is not supported. It has an implicit structure which allows, for example, text files stored in UNIX format (lines terminated by LF) to be converted to a format in which lines are terminated by CR followed by LF and vice versa.

Constraint Set Name:

{ISO standard 8571 constraint set name(5) unstructured(1)}

Abstract Syntax:

The abstract syntax of each data element is an instance of the data type 8859String.

Abstract Syntax Name:

{ISO registration-authority NBS FTAM( ) abstract syntax(2) NBS-AS1(0)}

Transfer Syntax:

An implementation supporting this abstract syntax shall support a transfer syntax for each data element obtained by applying the ASN.1 Basic Encoding Rules (ISO 8825) to the data elements and concatenating the resulting octets.

Note: this transfer syntax is not self-delimiting.

Implementations may optionally support other named transfer syntaxes for this abstract syntax.

Transfer Syntax Name:

{ISO registration-authority NBS FTAM( ) transfer syntax(3) NBS-TS1(0)}

Concatenation:

Concatenation of a document of this type with another document of this type is possible and produces a document of the same type consisting of one data unit which is the concatenation of the octetstring(s) of one file with the octetstring(s) of the other file.

Note: The boundary of the original OctetString is no longer visible.

Simplification:

A document of type NBS-3 can be accessed as a document of type NBS-1 by specifying a document type of NBS-1 in the Contents Type parameter of the F-OPENrequest, and limiting access context to US on F-READ.

Entry Number: NBS-4

Document_Type_Name:

{ISO registration-authority NBS FTAM( ) document(6) Text(3) parameter}

Note: "parameter" is a parameter which will be appended to the registered identifier in an OBJECT IDENTIFIER.

Document_Descriptor Value: Sequential Text File

Document_Semantics:

The document consists of an unbounded series of data units. Each data unit contains one data element which is a character string. Each character is taken from the IA5 character set.

Scope and Field of_Application:

The document type defines the contents of a file for storage and for transfer using FTAM.

Note: storage refers to apparent storage within the virtual filestore.

Constraint_Set_Name:

{ISO standard 8571 constraint set name(5) sequential flat(2)}

Additional_Constraints:

FADU Identity will be limited to begin, end, first and next.

Abstract_Syntax:

The abstract syntax of each data element is an instance of the data type IA5String.

The abstract syntax of each data unit is specified by the parameter.

Abstract_Syntax Name:

{ISO registration-authority NBS FTAM( ) abstract syntax(2) NBS-AS1(0)}

Transfer_Syntax:

An implementation supporting this abstract syntax shall support a transfer syntax for each data element obtained by applying the ASN.1 Basic Encoding Rules (ISO 8825) to the data elements and concatenating the resulting octets.

Note: This transfer syntax is not self-delimiting.

Implementations may optionally support other named transfer syntaxes for this abstract syntax.

Transfer_Syntax Name:

{ISO registration-authority NBS FTAM( ) transfer syntax(3) NBS-TS1(0)}

Concatenation:

Concatenation of a document of this type with another document of this type is possible and produces a document of the same type consisting of a series of IA5Strings which is the result of placing the series of IA5Strings from one file of this type after the last IA5String in the original file.

Note: The boundary of the original sequence is no longer visible.

Simplification:

A document of type NBS-4 can be accessed as a document of type NBS-1 by specifying a document type of NBS-1 in the Contents Type parameter on the F-OPENrequest, and limiting access context to UA on F-READ.

Entry Number: NBS-5

Document Type Name:

{ISO registration-authority NBS FTAM( ) document(6) 8859Text(4) parameter}

Note: "parameter" is a parameter which will be appended to the registered identifier in an Object Identifier.

Document Descriptor Value: Sequential Text File

Document Semantics:

The document consists of an unbounded series of data units. Each data unit contains one data element which is a character string. Each character is taken from the ISO 8859/1 character set.

Scope and Field of Application:

The document type defines the contents of a file for storage and transfer using FTAM.

Note: Storage refers to apparent storage within the virtual filestore.

Constraint Set Name:

{ISO standard 8571 constraint set name(5) sequential flat(2)}

Additional Constraints:

FADU Identity will be limited to begin, end, first and next.

Abstract Syntax:

The abstract syntax of each data element is an instance of the data type 8859String.

The abstract syntax of each data unit is specified by the parameter.

Abstract Syntax Name:

{ISO registration-authority NBS FTAM( ) abstract syntax(2) NBS-AS1(0)}

Transfer Syntax:

An implementation supporting this abstract syntax shall support a transfer syntax for each data element obtained by applying the ASN.1 Basic Encoding Rules (ISO 8825) to the data elements and concatenating the resulting octets.

Note: This transfer syntax is not self-delimiting.

Implementations may optionally support other named transfer syntaxes for this abstract syntax.

Transfer Syntax Name:

{ISO registration-authority NBS FTAM( ) transfer syntax(3) NBS-TS1(0)}

Concatenation:

Concatenation of a document of this type with another document of this type is possible and produces a document of the same type consisting of a series of 8859Strings which is the result of placing the series of 8859Strings from one file immediately following the last 8859String in the original file.

Note: The boundary of the original series is no longer visible.

Simplification:

A document of type NBS-5 can be accessed as a document of type NBS-1 by specifying a document type of NBS-1 in the Contents Type parameter on the F-OPENrequest, and limiting access context to UA on F-READ.

Entry Number: NBS-6

Document Type Name:

{ISO registration-authority NBS FTAM( ) document(6) SEQUENTIAL(5) parameter}

Note - "parameter" is a parameter which will be appended to the registered identifier in an Object Identifier.

Document Descriptor Value: Sequential File

Document Semantics:

The document consists of an unbounded series of data units. Each data unit contains an unbounded series of data elements. Each data element is a data type from the set of primitive data types defined in the main body of this document. Each data unit contains the same data element types in the same order as all other data units.

Scope and Field of Application:

The document type defines the contents of a file for storage and transfer using FTAM.

Note: Storage refers to apparent storage within the virtual filestore.

Constraint Set Name:

{ISO standard 8571 constraint set name(5) sequential flat(2)}

Additional Constraints:

FADU Identity will be limited to begin, end, first, and next.

Abstract Syntax:

The abstract syntax of each data element is an instance of one of the primitive data types defined in the main body of this document.

The Abstract syntax of each data unit is specified by the parameter.

Abstract Syntax Name:

{ISO registration-authority NBS FTAM( ) abstract syntax(2) NBS-AS1(0)}
optionally, {ISO registration-authority NBS FTAM( ) abstract syntax(2)
NBS-AS2(1)}

Transfer Syntax:

An implementation supporting this abstract syntax shall support a transfer syntax for each data element obtained by applying the ASN.1 Basic Encoding Rules (ISO 8825) to the data elements and concatenating the resulting octets.

Note: This transfer syntax is not self-delimiting.

Implementations may optionally support other named transfer syntaxes for this abstract syntax.

Transfer Syntax Name:

{ISO registration-authority NBS FTAM( ) transfer syntax(3) NBS-TS1(0)}

Concatenation:

Concatenation of a document of this type with another document of this type is possible and produces a document of the same type consisting of a series of data units which is the result of placing the series of data units from one file immediately following the last data unit of the original file.

Note: The boundary of the original file is no longer visible.

Simplification:

A document of type NBS-6 can be accessed as a document of type NBS-1 by specifying a document type of NBS-1 in the Contents Type parameter in the F-OPENrequest, and limiting access context to UA on F-READ.

Entry Number: NBS-7

Document Type Name:

{ISO registration-authority NBS FTAM ( ) document(6) RANDOM(6) parameter}

Note: "parameter" is a parameter which will be appended to the registered identifier in an Object Identifier.

Document Descriptor Value: Random Access File

Document Semantics:

The document consists of an unbounded series of data units. Each data unit contains an unbounded series of data elements. Each data element is a data type from the set of primitive data types defined in the main body of this document. Each data unit contains the same data types in the same order as all other data units in the file.

Scope and Field of Application:

The document type defines the contents of a file for storage and transfer using FTAM.

Note: storage refers to apparent storage within the virtual filestore.

Constraint Set name:

{ISO registration-authority NBS FTAM( ) constraint set name(5)
NBS Ordered Flat(2)}

Abstract Syntax:

The abstract syntax of each data element is an instance of one of the primitive data types defined in the main body of this document.

The Abstract syntax of each data unit is specified by the parameter.

Abstract Syntax name:

{ISO registration-authority NBS FTAM( ) abstract syntax(2) NBS-AS1(0)}
optionally, {ISO registration-authority NBS FTAM( ) abstract syntax(2)
NBS-AS2(1)}

Transfer Syntax:

An implementation supporting this abstract syntax shall support a transfer syntax for each data element obtained by applying the ASN.1 Basic Encoding Rules (ISO 8825) to the data element and concatenating the resulting octets.

Note: This transfer syntax is not self delimiting.

Implementations may optionally support other named transfer syntaxes for this abstract syntax.

Transfer Syntax Name:

{ISO registration-authority NBS FTAM( ) transfer syntax(3) NBS-TS1(0)}

Concatenation:

Concatenation of a document of this type with another document of this type is possible and produces a document of the same type consisting of a series of data units which is the result of placing the series of data units from one file immediately following the last data unit of the original file.

Note: the boundary of the original file is no longer visible.

Simplification:

A document of type NBS-7 can be accessed as a document of type NBS-1 by specifying a document type of NBS-1 in the Contents Type parameter of the F-OPENrequest, and limiting access context to UA on F-READ.

A document of type NBS-7 can be accessed as a document of type NBS-6 by specifying a document type of NBS-6 in the Contents Type parameter of the F-OPENrequest.

Entry Number: NBS-8

Document Type Name:

{ISO registration-authority NBS FTAM( ) document(6) INDEXED(7) p1 p2}

Note: "p1" and "p2" are parameters which will be appended to the registered identifier in an Object Identifier.

Document Descriptor Value: Indexed Sequential File

Document Semantics:

The document consists of an unbounded series of data units. Each data unit is an unbounded series of data elements. Each data element is a data type from the set of primitive data types defined in the main body of this document. Each data unit contains the same data types in the same order as all other data units in the file.

Each data unit in the file has a key associated with it. The key of each data unit is of the same data type as the key of all other data units in the file and is a single data element from the set of primitive data types defined in the main body of this document.

Scope and Field of Application:

The document type defines the contents of a file for storage and for transfer using FTAM.

Constraint Set Name:

{ISO registration-authority NBS FTAM( ) constraint set name(2)
Indexed Flat(1)}

Abstract Syntax:

The abstract syntax of each data element is an instance of one of the primitive data types defined in the main body of this document.

The Abstract syntax of each data unit is specified by the parameter p1.

The Abstract syntax of the data unit key (FADU Identifier) is specified by the parameter p2.

Abstract Syntax Name:

{ISO registration-authority NBS FTAM( ) abstract syntax(2) NBS-AS1(0)}
optionally, {ISO registration-authority NBS FTAM( ) abstract syntax (2)
NBS-AS2(1)}

Transfer Syntax:

An implementation supporting this abstract syntax shall support a transfer syntax for each data element obtained by applying the ASN.1 Basic Encoding Rules (ISO 8825) to the data elements and concatenating the resulting octets.

Note: this transfer syntax is not self-delimiting.

Implementations may optionally support other named transfer syntaxes for this abstract syntax.

Transfer Syntax Name:

{ISO registration-authority NBS FTAM( ) transfer syntax(3) NBS-TS1(0)}

Concatenation:

A document of this type may not be concatenated with a document of this type or any other type.

Simplification:

A document of type NBS-8 can be accessed as a document of type NBS-1 by specifying document type NBS-1 in the Contents Type parameter on the F-OPENrequest, and limiting access context to UA on F-READ.

A document of type NBS-8 can be accessed as a document of type NBS-6 by specifying document type NBS-6 in the Contents Type parameter on the F-OPENrequest.

A document of type NBS-8 can be accessed as a document of type NBS-7 by specifying document type NBS-7 in the Contents Type parameter on the F-OPENrequest.

Entry Number: NBS-9

Document_Type Name:

{ISO registration-authority NBS FTAM( ) document(6) FILE_DIRECTORY(8)}

Document_Descriptor Value: FileDirectory File

Document_Semantics:

The document consists of an unbounded sequence of data units. Each data unit consists of one and only one data element of type FileDirectoryEntry (a complex data type defined in the main body of this document).

Scope and Field of_Application:

This document defines the contents of a file for transfer (not for storage) using FTAM.

Constraint_Set_Name:

{ISO registration-authority NBS FTAM( ) constraint set name(5) Sequential Flat(1)}

Additional_Constraints:

FileDirectory Files may be Selected, Opened, Read, Closed, Created, and Deleted. They may not be Written or Modified (except as a side-effect of actions performed on individual files contained within a FileDirectory). DataUnits within a FileDirectory may only be accessed sequentially.

Abstract_Syntax:

An indefinite series of data units. Each data unit contains one data element of type FileDirectoryEntry. Each data element consists of a required FileName and a number of optional Attributes.

Abstract_Syntax Name:

{ISO registration-authority NBS FTAM( ) abstract syntax(3) filedirectory
entry(0)}

Transfer_Syntax:

An implementation supporting this data type shall support a transfer syntax for each data value obtained by applying ASN.1 Basic Encoding Rules to the data type FileDirectoryEntry in the data value and concatenating the resulting octets.

Note: this transfer syntax is not self-delimiting.

Implementations may also support other named transfer syntaxes for this abstract syntax.

Transfer Syntax Name:

{ISO registration-authority NBS FTAM( ) transfer syntax(3) NBS-TS1(0)}

Concatenation:

A document of this type cannot be concatenated with a document of this type or any other type.

Simplification:

A document of this type cannot be simplified.

Part 2: Constraint Sets

Constaint Set Title: NBS-Ordered Flat

Constraint Set Name:

{ISO registration-authority NBS FTAM( ) constraint set name(5)
NBS Ordered Flat(2)}

Field of Application: Files which are structured into a sequence of individual FADUs and to which access may be made on a FADU basis by position in the sequence.

Node Names: none

Actions: Locate, Read, Replace, Insert, Erase

Special Action Parameters: none

Special Action Semantics:

Erase: Used on the root node to empty the file. When used on a leaf node, it leaves a FADU with no associated data unit.

Insert: Allowed only at end of file. The new node is inserted following all existing nodes in the file or on a leaf node with no existing data unit. The inserted data unit is associated with the currently existing leaf node.

Available Access Contexts: HA, FA, UA, US

Erase and Locate Context: HA

Constraints on Structure: The root node shall not have an associated data unit. All children of the root node shall be leaf nodes and may have an associated data unit. All arcs from the root node shall be of length one.

Creation State: Root node without an associated data unit.

FADU Identity: begin, end, first, last,
current, next, previous,
traversal number (greater than or equal to one)

Location After Open: root node

Beginning of File: root node

End of File: No node is selected. Previous gives the last node in the traversal sequence, current and next result in an error.

Constraint Set Title: Indexed Flat

Constraint Set Name:

{ISO registration-authority NBS FTAM( ) constraint set name(5)
NBS Indexed Flat(1)} .

Field of Application: This constraint set is for representing single key ISAM files where the keys are the FADU identifiers for the leaf nodes. The keys are restricted to being single primitive data types, and restricted to all keys being of the same primitive data type.

Node Names: Any single primitive data type.

Actions: Locate, Read, Replace, Insert, Erase

Special Action Parameters: none

Special Action Semantics: Locate: The specified FADU is made the current FADU. If the FADU Id form is used, the least recently inserted FADU with the specified FADU at level 1 is located.

Read: Allowed at root and leaves. If there is another FADU after (in pre-order traversal order) the one read with the same FADU Id, a diagnostic on TRANSFER_END will indicate this fact.

Insert: Insert the specified FADU (level 1 only) in the lexical order of the key primitive data type. If there is already another FADU with the specified FADU Id, insert the new one after (in pre-order traversal) the existing FADUs and indicate that this was done via a diagnostic on TRANSFER_END.

Replace: Allowed only at leaves and only in access context US (DU only w/o delimiters). Only allowed with write operation of "current" (i.e., preceded by locate) or "Previous" (i.e., preceded by read).

Erase: If the addressed FADU is the root, the file is reduced to the initial state.

Available Access Contexts: HA, FA, UA, US

Erase and Locate Context: HA

Constraints on Structure: The root node shall not have an associated data unit or/and FADU Id. All children of the root node shall be leaf nodes and shall have an associated data unit and FADU Id. All arcs from the root node shall be of length one. Some primitive types may not be supported as keys.

Creation State: Root node without an associated data unit or FADU Id.

FADU Identity: begin, end,
current, next, previous,
FADU Id

Location After Open: root node

Beginning of File: root node

End of File: No node is selected. Previous gives the last node in the traversal sequence, current and next result in an error.

Part 3: Abstract Syntaxes

Abstract Syntax: NBS-AS1

Abstract Syntax Name:

{ISO registration-authority NBS FTAM( ) Abstract Syntax(2) Basic(0)}

Abstract Syntax Definition:

```
DE ::=Choice{INTEGER,
             BOOLEAN,
             IA5String,
             8859String,
             OCTETSTRING,
             UniversalTime,
             GeneralizedTime,
             Null}
8859String ::= [PRIVATE 1] Implicit 8859CharacterString
Transfer syntax name:      -- 8859CharacterString is a string of characters from
                           -- the ISO 8859 character set
{ISO registration-authority NBS FTAM( ) Transfer Syntax(4) NBS-TS1 (0)}
```

Abstract Syntax: NBS-AS2

Abstract Syntax Name:

{ISO registration-authority NBS FTAM( ) abstract syntax(2) FloatingPoint(1)}

Abstract Syntax Definition:

```
    FloatingPointNumber ::= [PRIVATE 0] CHOICE
           {
              finite [0] IMPLICIT SEQUENCE
              {
                        Sign,
                        mantissa BITSTRING,
                        exponent INTEGER
                    },
               infinity [1] IMPLICIT Sign,
               signalling NaN [2] Implicit NaN,
               quietNaN [3] IMPLICIT NaN,
               zero [4] IMPLICIT NULL
            }
     Sign ::= INTEGER ({positive(0), negative(1)}

     NaN ::= INTEGER
```

Transfer Syntax Name:

{ISO registration-authority NBS FTAM( ) transfer syntax(3) NBS-TS1(0)}

Abstract Syntax: NBS-AS3

Abstract Syntax Name:

{ISO registration-authority NBS FTAM( ) abstract syntax(2) FileDirectoryElement(2)}

Abstract Syntax Definition:

```
FileDirectoryEntry ::= [PRIVATE 2] IMPLICIT SEQUENCE{
                                  fileName GraphicString,
                                  ContentsType}

ContentsType ::= CHOICE
                document-type-Name[0] IMPLICIT OBJECT IDENTIFIER,
                constraint-set-and-abstract-syntax [1] IMPLICIT SEQUENCE{
                           constraint-set-Name[0] IMPLICIT OBJECT IDENTIFIER,
                           abstract-syntax-Name[1] IMPLICIT OBJECT IDENTIFIER}
```

Transfer Syntax Name:

{ISO registration-authority NBS FTAM( ) transfer syntax(3) NBS-TS1(0)}

Part 4: Transfer Syntaxes

Transfer Syntax: NBS-TS1

Transfer Syntax Name:

{ISO registration-authority NBS FTAM( ) transfer syntax(3) NBS-TS1(0)}

Encoding Rules:

ASN.1 Basic Encoding Rules shall apply

The first bit of a "fraction" must be "1"

Transfer Syntax Definition:

The transfer syntax shall be that which results from applying the encoding rules described above to the individual data elements.

ADDENDUM 1

Note on FTAM and X.400 Character Sets

On July 21, 1986 a group of twelve individuals from the FTAM and X.400 SIGs met to resolve differences in recommended use of character sets. The following was agreed (in favor, 9; opposed, 2; abstaining, 1) by these individuals:

"Both SIGs should implement IA5 for the current phase of development, and independently support expanded character sets. It is recommended that the FTAM and X.400 SIGs support both 8859/1 and 6937/2 in the next phase of their agreements."

Neither SIG brought forward to the plenary on Thursday July 24, 1986 a recommendation on this issue. However, it was raised for plenary discussion. The plenary felt (in favor, 22; opposed, 3; abstaining, 1) that this information should be carried in some form in this document in addition to inclusion in the minutes. Hence, it is included as an addendum so as to keep the information associated with this document while showing that it has not been accepted for inclusion in the main body of this document.

You will receive the documents from the next workshop by either attending the workshop or completing and returning the form below.

READER RESPONSE FORM

Please retain my name for the next mailing of the NBS/OSI Implementors Workshop

NAME ____

ADDRESS ____

____

____

PHONE NO. ____

Mail this page to: Kim Brink
National Bureau of Standards
Bldg. 225/B217
Gaithersburg, MD 20899

NBS-114A (REV. 2-8C)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET *(See instructions)*	1. PUBLICATION OR REPORT NO. NBSIR 86-3385-1	2. Performing Organ. Report No.	3. Publication Date JULY 1986

4. TITLE AND SUBTITLE

Implementation Agreements Among Implementors of OSI Protocols

5. AUTHOR(S)

John Heafner, Editor

6. PERFORMING ORGANIZATION *(If joint or other than NBS, see instructions)*

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
Gaithersburg, MD 20899

7. Contract/Grant No.

8. Type of Report & Period Covered

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS *(Street, City, State, ZIP)*

10. SUPPLEMENTARY NOTES

□ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)*

This document records current agreements on implementation details of Open Systems Interconnection protocols among the organizations participating in the NBS/OSI Workshop Series for Implementors of OSI Protocols. These decisions are documented to facilitate organizations in their understanding of the status of agreements. This is a standing document that is updated after each workshop (about every 2-1/2 months). A reference list of standards and a list of contributing organizations are included in the Appendix.

12. KEY WORDS *(Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)*

local area networks; NBS/OSI Workshop; network protocols; Open systems interconnection; OSINET; testing protocols

13. AVAILABILITY

□ Unlimited
☒ For Official Distribution. Do Not Release to NTIS
□ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.
□ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES

15. Price